UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Monitoring and Diagnosis of Process Systems Using Kernel-Based Learning Methods

by

Gorden Takawadiyi Jemwa

Dissertation presented for the Degree of

DOCTOR OF PHILOSOPHY IN ENGINEERING

in the Department of Process Engineering at the University of Stellenbosch

Promotor: Prof. Chris Aldrich

Stellenbosch
March 2007

Declaration

I, the undersigned, hereby declare that the work contained in this dissertation is my own original work and has not previously in its entirety or in part been submitted at any university for a degree.

Signature: _____

Date: _____

Synopsis

The development of advanced methods of process monitoring, diagnosis, and control has been identified as a major 21st century challenge in control systems research and application. This is particularly the case for chemical and metallurgical operations owing to the lack of expressive fundamental models as well as the nonlinear nature of most process systems, which makes established linearization methods unsuitable. As a result, efforts have been directed in the search of alternative approaches that do not require fundamental or analytical models. Data-based methods provide a very promising alternative in this regard, given the huge volumes of data being collected in modern process operations as well as advances in both theoretical and practical aspects of extracting information from observations.

In this thesis, the use of kernel-based learning methods in fault detection and diagnosis of complex processes is considered. Kernel-based machine learning methods are a robust family of algorithms founded on insights from statistical learning theory. Instead of estimating a decision function on the basis of minimizing the training error as other learning algorithms, kernel methods use a criterion called large margin maximization to estimate a linear learning rule on data embedded in a suitable feature space. The embedding is implicitly defined by the choice of a kernel function and corresponds to inducing a nonlinear learning rule in the original measurement space. Large margin maximization corresponds to developing an algorithm with theoretical guarantees on how well it will perform on unseen data.

In the first contribution, the characterization of time series data from process plants is investigated. Whereas complex processes are difficult to model from first principles, they can be identified using historic process time series data and a suitable model structure. However, prior to fitting such a model, it is important to establish whether the time series data justify the selected model structure. Singular spectrum analysis (SSA) has been used for time series identification. A nonlinear extension of SSA is proposed for classification of time series. Using benchmark systems, the proposed extension is shown to perform better than linear SSA. Moreover, the method is shown to be useful for filtering noise in time series data and, therefore, has potential applications in other tasks such as data rectification and gross error detection.

Multivariate statistical process monitoring methods are well-established techniques for efficient information extraction from multivariate data. Such information is usually compact and amenable to graphical representation in two or three dimensional plots. For process monitoring purposes control limits are also plotted on these charts. These control limits

are usually based on a hypothesized analytical distribution, typically the Gaussian normal distribution. A robust approach for estimating confidence bounds using the reference data is proposed. The method is based on one-class classification methods. The usefulness of using data to define a confidence bound in reducing fault detection errors is illustrated using plant data.

The use of both linear and nonlinear supervised feature extraction is also investigated. The advantages of supervised feature extraction using kernel methods are highlighted via illustrative case studies. A general strategy for fault detection and diagnosis is proposed that integrates feature extraction methods, fault identification, and different methods to estimate confidence bounds. For kernel-based approaches, the general framework allows for interpretation of the results in the input space instead of the feature space.

An important step in process monitoring is identifying a variable responsible for a fault. Although all faults that can occur at any plant cannot be known beforehand, it is possible to use knowledge of previous faults or simulations to anticipate their recurrence. A framework for fault diagnosis using one-class support vector machine (SVM) classification is proposed. Compared to other previously studied techniques, the one-class SVM approach is shown to have generally better robustness and performance characteristics.

Most methods for process monitoring make little use of data collected under normal operating conditions, whereas most quality issues in process plants are known to occur when the process is "in-control". In the final contribution, a methodology for continuous optimization of process performance is proposed that combines support vector learning with decision trees. The methodology is based on continuous search for quality improvements by challenging the normal operating condition regions established via statistical control. Simulated and plant data are used to illustrate the approach.

Oorsig

Die ontwikkeling van gevorderde metodes van prosesmonitering, diagnose en -beheer is geïdentifiseer as 'n groot 21ste eeuse uitdaging in die navorsing en toepassing van beheerstelsels. Dit is veral die geval in die chemiese en metallurgiese bedryf, a.g.v. die gebrek aan fundamentele modelle, sowel as die nielineêre aard van meeste prosesstelsels, wat gevestigde benaderings tot linearisasie ongeskik maak. Die gevolg is dat pogings aangewend word om te soek na alternatiewe benaderings wat nie fundamentele of analitiese modelle benodig nie. Data-gebaseerde metodes voorsien belowende alternatiewe in dié verband, gegewe die enorme volumes data wat in moderne prosesaanlegte geberg word, sowel as die vooruitgang wat gemaak word in beide die teoretiese en praktiese aspekte van die onttrekking van inligting uit waarnemings.

In die tesis word die gebruik van kern-gebaseerde metodes vir foutopsporing en -diagnose van komplekse prosesse beskou. Kern-gebaseerde masjienleermetodes is 'n robuuste familie van metodes gefundeer op insigte uit statistiese leerteorie. Instede daarvan om 'n besluitnemingsfunksie te beraam deur passingsfoute op verwysingsdata te minimeer, soos wat gedoen word met ander leermetodes, gebruik kern-metodes 'n kriterium genaamd groot marge maksimering om lineêre reëls te pas op data wat ingebed is in 'n geskikte kenmerkruimte. Die inbedding word implisiet gedefinieer deur die keuse van die kern-funksie en stem ooreen met die indusering van 'n nielineêre reël in die oorspronklike meetruimte. Groot marge-maksimering stem ooreen met die ontwikkeling van algoritmes waarvan die prestasie t.o.v. die passing van nuwe data teoreties gewaarborg is.

In die eerste bydrae word die karakterisering van tydreeksdata van prosesaanlegte ondersoek. Alhoewel komplekse prosesse moeilik is om vanaf eerste beginsels te modelleer, kan hulle geïdentifiseer word uit historiese tydreeksdata en geskikte modelstrukture. Voor so 'n model gepas word, is dit belangrik om vas te stel of die tydreeksdata wel die geselekteerde modelstruktuur ondersteun. 'n Nielineêre uitbreiding van singuliere spektrale analise (SSA) is voorgestel vir die klassifikasie van tydreekse. Deur gebruik te maak van geykte stelsels, is aangetoon dat die voorgestelde uitbreiding beter presteer as lineêre SSA. Tewens, daar word ook aangetoon dat die metode nuttig is vir die verwydering van geraas in tydreeksdata en daarom ook potensiële toepassings het in ander take, soos datarektifikasie en die opsporing van sistematiese foute in data.

Meerveranderlike statistiese prosesmonitering is goed gevestig vir die doeltreffende onttrekking van inligting uit meerveranderlike data. Sulke inligting is gewoonlik kompak en geskik vir voorstelling in twee- of drie-dimensionele grafieke. Vir die doeleindes van prosesmonitering word beheerlimiete dikwels op sulke grafieke aangestip. Hierdie beheerlimi-

ete word gewoonlik gebaseer op 'n hipotetiese analitiese verspreiding van die data, tipiese gebaseer op 'n Gaussiaanse model. 'n Robuuste benadering vir die beraming van betroubaarheidslimiete gebaseer op verwysingsdata, word in die tesis voorgestel. Die metode is gebaseer op eenklas-klassifikasie en die nut daarvan deur data te gebruik om die betroubaarheidsgrense te beraam ten einde foutopsporing te optimeer, word geïllustreer aan die hand van aanlegdata.

Die gebruik van beide lineêre en nielineêre oorsiggedrewe kenmerkonttrekking is vervolgens ondersoek. Die voordele van oorsiggedrewe kenmerkonttrekking deur van kern-metodes gebruik te maak is beklemtoon deur middel van illustratiewe gevallestudies. 'n Algemene strategie vir foutopsporing en -diagnose word voorgestel, wat kenmerkonttrekkingsmetodes, foutidentifikasie en verskillende metodes om betroubaarheidsgrense te beraam saamsnoer. Vir kern-gebaseerde metodes laat die algemene raamwerk toe dat die resultate in die invoerruimte vertolk kan word, in plaas van in die kenmerkruimte.

'n Belangrike stap in prosesmonitering is om veranderlikes te identifiseer wat verantwoordelik is vir foute. Alhoewel alle foute wat by 'n chemiese aanleg kan plaasvind, nie vooraf bekend kan wees nie, is dit moontlik om kennis van vorige foute of simulasies te gebruik om die herhaalde voorkoms van die foute te antisipeer. 'n Raamwerk vir foutdiagnose wat van eenklas-steunvektormasjiene (SVM) gebruik maak is voorgestel. Vergeleke met ander tegnieke wat voorheen bestudeer is, is aangetoon dat die eenklas-SVM benadering oor die algemeen beter robuustheid en prestasiekenmerke het.

Meeste metodes vir prosesmonitering maak min gebruik van data wat opgeneem is onder normale bedryfstoestande, alhoewel meeste kwaliteitsprobleme ondervind word waneer die proses "onder beheer" is. In die laaste bydrae, is 'n metodologie vir die kontinue optimering van prosesprestasie voorgestel, wat steunvektormasjiene en beslissingsbome kombineer. Die metodologie is gebaseer op die kontinue soeke na kwaliteitsverbeteringe deur die normale bedryfstoestandsgrense, soos bepaal deur statistiese beheer, te toets. Gesimuleerde en werklike aanlegdata is gebruik om die benadering te illustreer.

...To my parents.
A special dedication to the memory of my mother, Demetria Tafireyi (1947–2004)

Acknowledgments

I am highly indebted to my study leader Chris Aldrich, whose excellent knowledge and insights inspired the development of this dissertation in innumerable ways. Through a number of extended discussions in his office, Chris has consistently displayed an exemplary research standard and professionalism that I hold in great respectful admiration and will always aspire to in all my future pursuits.

A sincere appreciation to the Department of Process Engineering for providing an environment and support structure that facilitated the progress and successful completion of the work contained herewith. Heartfelt thanks to Juliana Steyl for being there for me since I started studying at the university. Thanks also to Lynette Bresler, Ina van Zyl, and Enid Thom for taking care of all the administrative issues.

I also would like to acknowledge discussions and assistance I have received from the following members of the machine learning research community; Gökhan Bakhir, Arthur Gretton, Alexandros Karatzoglou, Gert Lanckriet, Chih-Jen Lin, Alain Rakotomamonjy, Bernhard Schölkopf, Alex Smola, David M.J. Tax, Jason Weston. I have also benefited much through various machine learning blogs, particularly those of John Langford and Olivier Bousquet.

Although studying in a foreign country can be a very difficult enterprise, I have been very fortunate to have met very friendly people who made "sailing the rough seas" enjoyable and all the more worthwhile. I extend my inexpressible gratitude to my South African family – the Van der Merwe family (Willie Snr., Lettie, Willie Jnr., and Maria) who showered me with all the thoughts and love one can only find within a family. And to Lettie, "Herewith is the final exclamation mark! May all the nagging cease henceforth."

I am very grateful to members of different research groups whom I met and shared office space with during my studies. Their unwavering belief in me when I could not see the end as well as cherished lighter moments gave me strength to proceed. Many of these, in particular Wezi Banda, John Burchell and Paul Botha, became good friends even outside the boundaries of research cooperation.

Thank you to my brothers, sisters, family members and friends for their patience and understanding during the course of my studies.

Finally, a very special thank you to my parents who have been a source of inspiration as well as guiding and supporting me in all my endeavors since I was young. Without your love and encouragement, this would not have been possible.

This work was made possible through funding by the National Research Foundation (NRF), the Department of Process Engineering, and the University of Stellenbosch.

Contents

Nomenclature

$\boldsymbol{\alpha}$	vector of Lagrange α_i coefficients
$\boldsymbol{\phi}$	generalized mapping function
$(\mathbf{x} \cdot \mathbf{y})$	inner product between vectors $\mathbf{x}$ and $\mathbf{y}$ in Euclidean space
$\langle f, g \rangle$	inner product function between vector space of functions f and g in Hilbert space
$\ell(a, b)$	non-negative loss function between actual a and predicted b
$\mathcal{R}_{\text{emp}}$	empirical risk function
$\mathcal{H}$	Hilbert space – a generalized Euclidean space
$\mathbf{k}(\mathbf{x}, \mathbf{y})$	a kernel function evaluated on vectors $\mathbf{x}$ and $\mathbf{y}$
$\mathbb{N}$	set of natural numbers
$\mathbf{C}$	covariance matrix
$\mathbf{E}$	expected value
$\mathbf{I}$	identity square matrix
$\mathbf{K}$	kernel or Gram matrix where the $(i, j)^{\text{th}}$ entry is $\mathbf{K}_{ij} = \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	Gaussian distribution with mean $\boldsymbol{\mu}$ and standard deviation Σ
$\mathcal{T}$	training set
$\mathcal{U}(a, b)$	a continuous uniform distribution on the interval $[a, b]$
$\mathcal{X}$	input or observation space
$\mathcal{Y}$	output or target space
$\mathcal{Y}^{\mathcal{X}}$	the set of maps from $\mathcal{X}$ to $\mathcal{Y}$
$\mathscr{L}_D$	Lagrangian objective function in dual form

$\mathscr{L}_P$	Lagrangian objective function in primal form
$\|\cdot\|$	ℓ_2–norm
$\|\cdot\|_p$	ℓ_p–norm, $p \in \mathbb{N}$
$\mathcal{P}(\cdot)$	probability distribution function
$\mathbb{R}, \mathbb{R}^+$	set of real numbers and positive real numbers, respectively
$\mathcal{R}$	risk function
$\mathbf{w}$	weight vector
d	dimensionality of input space
h_d	VC (shattering) dimension
$I(\cdot)$	indicator function
m	size of training set
CUSUM	cumulative sum
ERM	empirical risk minimization
EWMA	exponentially weighted moving average
i.i.d.	independent and identically distributed
KFD	kernel Fisher discriminant analysis
KPCA	kernel principal component analysis
LDA	linear discriminant analysis
MC-SSA	Monte Carlo singular spectrum analysis
MLP	Multilayer perceptron
MSPC	multivariate statistical process control
MSPM	multivariate statistical process monitoring
PCA	principal component analysis
RKHS	reproducing kernel Hilbert space
SPE	squared prediction error
SRM	structural risk minimization
SSA	singular spectrum analysis

SVC	support vector classification
SVM	support vector machine
VC	Vapnik-Chervonenkis

Chapter 1

Introduction

> [For three centuries up to the end of the World War II], the model for technology was a mechanical one: the events that go on inside a star such as a sun [and] advance in technology meant–as it does in mechanical processes–more speed, higher temperatures, higher pressures. Since the end of World War II, however, the model of technology has become the biological process, the events inside an organism. And in an organism, processes are not organized around energy in the physicists's meaning of the term. They are organized around information.
>
> Peter F. Drucker, *Innovation and Entrepreneurship*

OVER the last few decades, there have been many socio-economic and technical developments that have seen important changes in the management and operation of industrial processes. For example, globalization challenges require companies to sustain and improve productivity while simultaneously meeting tighter quality specifications. Global warming and other threats to the earth's ecosystem have shifted attention toward sustainable economic activity. It is now imperative for industrial operations including chemical and metallurgical processes to comply with stricter safety and environmental regulatory constraints. As a result, companies are making huge investments in automating and integrating operator tasks as well as unit processes. However, plant control and supervisory tasks have also become more complex than can be solved by classical regulatory and statistical process control techniques. In light of these developments, achieving operational and business goals requires advanced control methodologies. Fortunately, advances in the information sciences have yielded data processing and analysis techniques that are very promising with respect to targeted applications in process control.

In this chapter the major factors impacting on control issues in process operations are highlighted. The basic fault detection and diagnostic framework is presented. Subsequently, the learning methodology is introduced, with emphasis on the desirable attributes

of handling finite and noisy data of unknown distribution as are observed in real process systems. Finally, the goals and scope of the thesis are outlined.

1.1 Trends in Process Systems Engineering

The changing social, economic and physical environment has witnessed important developments that are continuously placing greater demands on industrial processes. In the case of resource companies, for example, operations have to process ores with complex mineralogical compositions as reserves of less refractory ores become depleted. The opening up of the competitive space due to globalization is reducing market shares for most companies and a concomitant decrease in profit margins. Also, companies are now compelled to be responsive to varying customer demands yet still ensuring that product and process quality are sustained if not so much as improved. To maintain a competitive advantage, quality control management methodologies, like Six Sigma and ISO 9000, and other management programs have been developed to assist organizations in addressing some of these challenges.

Modern-day process operations have also become more complex owing to plant-wide integration and high-level automation of many process tasks. For example, recycling of process streams is now widely practiced to ensure efficient material and energy usage. Process plants have virtually become intricate information networks, with significant interactions among various subsystems and components. Although such interconnectivity facilitates the integration of operational tasks to achieve broader business strategic goals, it invariably complicates certain tasks like planning and scheduling, supervisory control and diagnosis of process operations.

Another factor that has affected process operations is the stringent regulatory framework aimed at minimizing risks posed by industrial activities to the environment. In addition, safety and health policies and practices are now priority issues in the modern-day plant. To this end, a number of systematic frameworks have been initiated, including process hazard analysis (PHA) and abnormal event management (AEM) and product life cycle management (PLM). PHA and AEM are aimed at ensuring process safety, while PLM places obligatory stewardship responsibilities to an organization throughout the life cycles of its entire product range, that is, from conception, through design and manufacture, service and disposal (Venkatasubramanian, 2005).

In response to these trends, plants are investing heavily in instrumentation to enable real time monitoring of process units and streams. New sensor technologies such as acoustic or vibrational signal monitoring and computer vision systems have been introduced in, among other, milling plants, multiphase processes, food processing, and combustion processes. Huge volumes of data are increasingly being generated, whereas the information content of these data is rarely enhanced. Moreover, some of the data obtained are not well-suited for analysis using classical approaches although containing useful information. This has motivated the need for advanced process control strategies that are knowledge-based and/or data-driven, collectively referred to as intelligent control systems.

Intelligent control systems are a group of methodologies aimed at exploiting information in plant data and experiential knowledge of plant operators (Åström and McAvoy, 1992;

McAvoy, 2002). The term "intelligent" refers to the automation of an engineering task without any implication to building a system capable of mimicking human intelligence as is sometimes incorrectly misunderstood. Process monitoring and diagnosis has developed from this perspective as an interdisciplinary field, with both theory and practice building on ideas from diverse fields such as fundamental process modelling, signal processing, statistics, machine learning, and other artificial intelligence-related areas. The development of advanced methods for monitoring, diagnosis, and control of abnormal events in complex systems and processes has been identified as an emerging major challenge in control systems research and application in the 21st century (Ogunnaike, 1996; Venkatasubramanian, 2005).

The intelligent systems framework should be viewed as a complementary rather than competing control methodology to classical or well-established techniques (Aldrich, 2000; Venkatasubramanian et al., 2003). By considering all perspectives a better understanding of complex systems can be achieved. Stephanopoulos and Han (1996) remarked that it is impossible to develop a generic, all-encompassing methodology for process fault diagnosis, since any model is only capable of explaining different facets of a very rich available knowledge resource. Therefore, a typical control and diagnostic system for an entire operation is integrative in form. In the control of nuclear-based processes, for example, different technologies, such as artificial neural networks, independent component analysis and wavelets, among other are combined in an overall control framework (Hines and Seibert, 2006).

1.2 Basic Fault Detection and Diagnosis Framework

A fault[(1)] is associated with any unacceptable anomalous behavior of components that causes a system to deviate from its normal operating condition, potentially leading to the overall failure of the system. Process plant faults can be classified according to their source: sensor faults which affect process measurement; actuator faults that cause changes in the actuators; process component faults arising from changes in process parameters and equipment; and faults induced by operator intervention. Further characterization is based on the time evolution of the fault: (i) abrupt or sudden faults; and (ii) incipient or slowly developing faults, for example equipment degradation or sensor drift. The main objectives in fault diagnosis are timely detection of aberrant process behavior, troubleshooting and eventual elimination of the root cause of the fault with little or no disruptions to the process operation.

Conceptually, the fault diagnosis problem involves building a model describing process behavior against which future evolution of the process is compared against. Potential faults in the system are then detected by monitoring the deviation of actual process behavior from expected behavior as predicted by a well-defined process model. An alarm is raised whenever the deviation, or a statistic derived from the deviations, violates pre-specified detection limits.

For optimal and reliable performance, any diagnostic system must be capable of producing a sufficient set of residuals that can capture the impact of faults of interest; this is closely related to a model's ability to capture regularity in the dynamic behavior of a system. In

(1) Note on terminology: In this thesis, the convention of Isermann and Ballé (1997) is adopted for definitions of the key terms used in monitoring, detection and diagnosis of processes. See Appendix A.

addition, a diagnostic system must also be capable of unique and robust inferencing of the origin of the fault (Stephanopoulos and Han, 1996). Since an observed variable measurement is a manifestation of the combined effects of different contributions including underlying physical and chemical phenomena, external disturbances, instrumentation disturbances, state and condition of equipment, and operator-induced actions, unique and robust fault identification requires decoupling the faults from the various contributions. This, in turn, requires existence of well-defined (explicit or implicit) relationships between observations (or symptoms) and a set of known failures. Such relationships are typically found by model fitting procedures on a set of observers. A model inversion can then be used to trace the likely root cause of a fault detected in a system. When the fault is completely different from what has been experienced or simulated, a diagnostic system should be capable of detecting and labeling these novel faults accordingly. Figure 1.1 shows a schematic outline of the fault detection and identification problem.

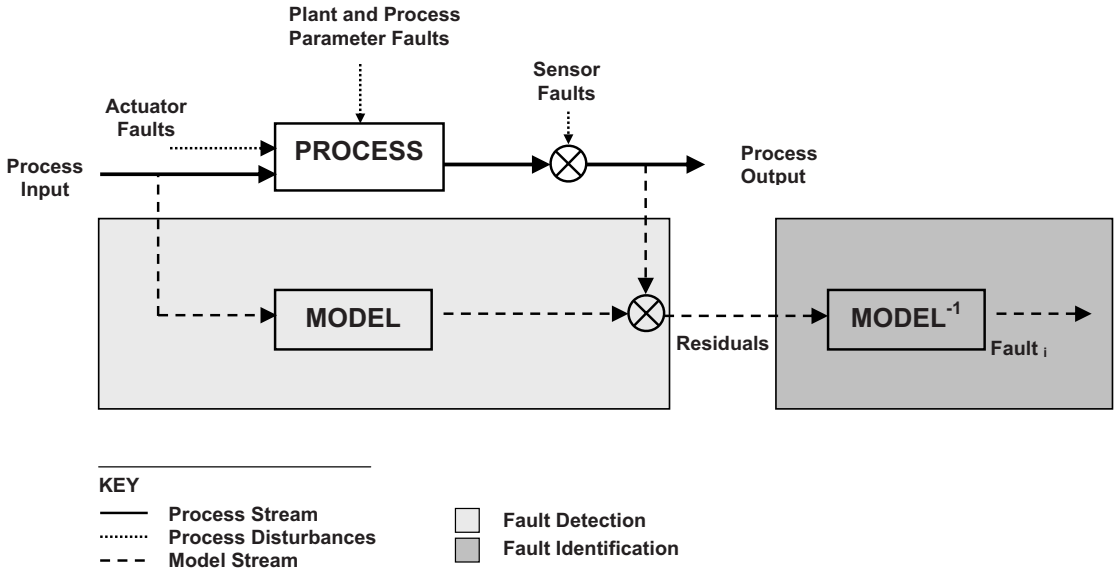

Figure 1.1: A basic outline of the fault diagnosis problem

Similar to other fields in scientific and engineering inquiry, mathematical models play an important role in process control. Successful monitoring and control is largely determined by how well a mathematical model is in capturing knowledge governing a system's behavior. Poor model accuracy degrades the performance of a diagnostic system. For real processes, an accurate model is not possible and, therefore, requirement of a model introduces another degree of freedom in the fault diagnosis framework, namely model uncertainty, in addition to the unknown disturbances. To guarantee reliability of a diagnostic system, the design and/or development of the process model needs careful consideration.

1.3 Process Model Selection in Fault Diagnosis

Classically, models have been derived analytically on the basis of fundamental physical relations. These first-principles or phenomenological models require extensive knowledge of

involved objects as well as the nature of interactions among them. Examples of fundamental models are diffusion models such as Fick's law used in describing transport processes in leaching. Unfortunately, complete knowledge of what exactly is taking place in real processes is often not available or very difficult to obtain. Hence, process models obtained do not account adequately for all observed behavior.

In other cases, particularly where measured data are not directly related to the underlying phenomena, it is not clear how to proceed in explaining the observed phenomena from a fundamental perspective. For example, in the froth flotation processes used in the recovery of metals, computer vision systems are increasingly being used for process monitoring (Aldrich et al., 1997; Hyötyniemi and Ylinen, 2000). A fundamental model for predicting the form of the froth surface requires knowledge of the interacting objects (such as the type and addition rates of reagents, ore mineralogy, grind size, as well as process parameters such as stirring rate, and air feed rate). Other examples include monitoring network activity, detecting fraudulent loan applications, use of acoustic signals in monitoring milling circuits, to name but a few. In all these cases, a fundamental model is difficult to define, at least not within the existing body of theoretical knowledge.

An alternative approach to fundamental modelling inspired by advances in information processing systems in the last half of the 20^{th} century is *learning from experience*, such as operator knowledge or process data. Knowledge-based systems use unstructured and fragmented or non-quantifiable information to develop a rule-based inferencing decision support system by means of symbolic reasoning. Of relevance to this study is learning from data, where the objective is to find a functional dependency that best explains the regularity or structure in process measurements with very few assumptions on the statistical nature of the governing mechanism.

Learning from experience can be considered a paradigm shift from classical scientific inquiry in which phenomena were explained in terms of materials within a well-defined metric system. Instead, problems are cast in terms of data representation, information and knowledge. Within this perspective, problems from diverse fields such as cognitive science, engineering, economics, genetics, medicine, and other fields where automated prediction is necessary converge when interpreted in terms of finding and analyzing relations in data. For example, upgrade of information content in biological data has also been identified as a dominant theme in computational biotechnology in the 21^{st} century (Stephanopolous, 1999), which is essentially similar to the process control perspective (Aldrich, 2000; Ogunnaike, 1996; Venkatasubramanian, 2005). Such information content upgrade can be achieved by statistical inferencing or planned experimental campaigns.

An alternative and suitable approach that uses little or no assumptions is machine learning. Machine learning is concerned with making machines or software programs that discover patterns in data by learning from examples. It brings together insights and tools of mathematics, theoretical and applied computational sciences, and statistics. In particular, it overlaps with many approaches that were proposed separately within the statistical community, for example decision trees (Breiman et al., 1993; Quinlan, 1986).

Machine learning algorithms for detecting anomalies, trends, or other "interesting" regularities or patterns have been proposed (Schölkopf and Smola, 2002). In the following section the statistical learning theory framework and the support vector machine (SVM) are intro-

duced, highlighting the desirable attributes of data-based function estimation when applied to practical problems such as are encountered in process systems.

1.4 Learning Theory: Main Insights and Algorithms

Generally, the problem of building models from data is not well-defined unless one is constrained to a specific subset of allowable models and a method exists for selecting the optimal model in the set. Given a set of finite data, statistical learning theory or Vapnik-Chervonenkis (VC) theory (Vapnik, 1998, 2000) prescribes a principled procedure for deciding which model to choose among the potentially infinite competing models with minimal assumptions on the structure of the model. This should be contrasted to the classical statistical approach in which a certain parametric form is specified beforehand, and fitting a model then consists of optimizing over the parameters.

Typically, parametric assumptions assume existence of a well-defined source generating distribution from which the observed data set have been sampled. As discussed later, the performance of parametric models is dependent on the size and quality of the available data used in fitting the model (Chapter 3). In particular, model quality degrades with increasing dimensionality of an observation vector because the number of samples required to obtain a proper fit increases exponentially with data dimensionality – a phenomenon commonly referred to as the "the curse of dimensionality" in statistics and empirical inference (e.g., Hastie et al. (2001)).

VC-theory provides a statistical learning bias that gives probability guarantees on the performance of a learning algorithm when presented with data not used in fitting the model. Kernel methods and particularly support vector machines are computationally feasible algorithms with good generalization bounds that are based on insights from statistical learning theory. The support vector machine (SVM) algorithm was the first practical machine learning tool to incorporate a VC-theory learning bias and a method for controlling the flexibility of the algorithm (Boser et al., 1992; Cortes and Vapnik, 1995). Briefly, an SVM learns a simple linear decision function after embedding the data into a potentially high-dimensional feature space. As explained in detail in Chapter 3, it is not necessary to explicitly perform the embedding into the feature space as long as the data objects only appear in terms of dot or inner products in the learning algorithm. The transformation is then implicitly determined by replacing the linear dot product with *kernel functions* in the computation. The use of kernel functions provides a simple yet elegant way to transform any linear algorithm into a nonlinear one. This insight has been used to yield nonlinear extensions of well-established statistical techniques such as principal component analysis (Schölkopf et al., 1998), discriminant analysis (Baudat and Anouar, 2000; Mika et al., 1999), canonical correlation analysis and independent component analysis (Bach and Jordan, 2002). This family of learning algorithms is generally referred to as kernel-based algorithms or kernel methods (Schölkopf and Smola, 2002).

Thus, SVMs (and kernel methods in general) possess the desirable properties of simple statistical complexity of linear models and very rich expressive capacity induced by the choice of the kernel function. Numerically, the SVM algorithm has a unique global solution, which contrasts favorably with other approaches such as artificial neural networks. Another

important property is the ability to learn a function for arbitrary distributions as well as outliers in the data. Moreover, SVMs can handle non-vectorial data such as strings, text, and graphs, which is not possible with most learning algorithms that can only work with vectorial data.

Because of these properties, they have found wide applications in fields such as bioinformatics, text mining, graphical modelling, and image processing, whereas applications in process systems engineering have been limited. Given the nature of process data (nonlinear and noisy) and prevailing challenges facing process industries, it is reasonable to extend the use of these methods to process systems.

1.5 Problem Statement and Objectives of the Study

Many process systems are generally characterized by complex nonlinear behavior that is difficult to model using fundamental approaches. Owing to lack of expressive first-principles model, attention is increasingly being focused on alternative approaches. A promising approach is using data-driven methods, especially given the huge volumes of data being generated on modern day process plants. However, data sampled from physical systems such as process plants is invariably corrupted by measurement and random errors, highly correlated and of limited size, which can pose problems when used within a purely statistical framework or other well-established techniques. There is growing emphasis on the use of computational-based technologies in process operations. These have found applications in modelling and automation of engineering tasks as well as providing a basis for developing an intelligent information system for

- monitoring and analysis;
- detecting abnormal operating conditions;
- identification of process trends; and
- planning and scheduling of plant recovery after a fault has been detected and eliminated.

Kernel-based approaches are a recent computational learning innovation that provide a useful methodology that can be used in data-driven process monitoring, analysis, and diagnosis tasks, particularly in exploiting redundancy in finite high-dimensional sampled data from nonlinear systems. In this thesis the use of kernel-based learning algorithms for developing new diagnostic methodologies for process systems applications is investigated. Specifically, the following objectives are addressed;

- Improving process identification; more specifically, advancing nonlinearity tests in time series signals obtained from dynamic processes;
- Detecting anomalous behavior and/or process drifts;
- Comparative analysis of latent variable projective methods, and
- Process optimization through continuous improvement opportunities aimed at reducing “in-control” or “common-cause” variation.

1.6 Scope of Thesis

Figure 1.2 shows a schematic high-level outline of the specific aspects addressed in this thesis within the broader framework of monitoring and diagnosis of technical processes. Note that while some overlap exists in the modularized functional tasks, the tasks as addressed at any level are generally different and context specific. For example, outlier detection in data validation may refer to removal of data points identified as arising from, say, a malfunctioning sensor. Alternatively, a drift in the sensor can be identified by an outlier detection model previously calibrated using data collected under normal operating process conditions. As might be expected, detecting malfunctioning sensor readings depends on the availability of specific knowledge of status of the process equipment at a point in time while sensor reading drift must be identified on the basis of data.

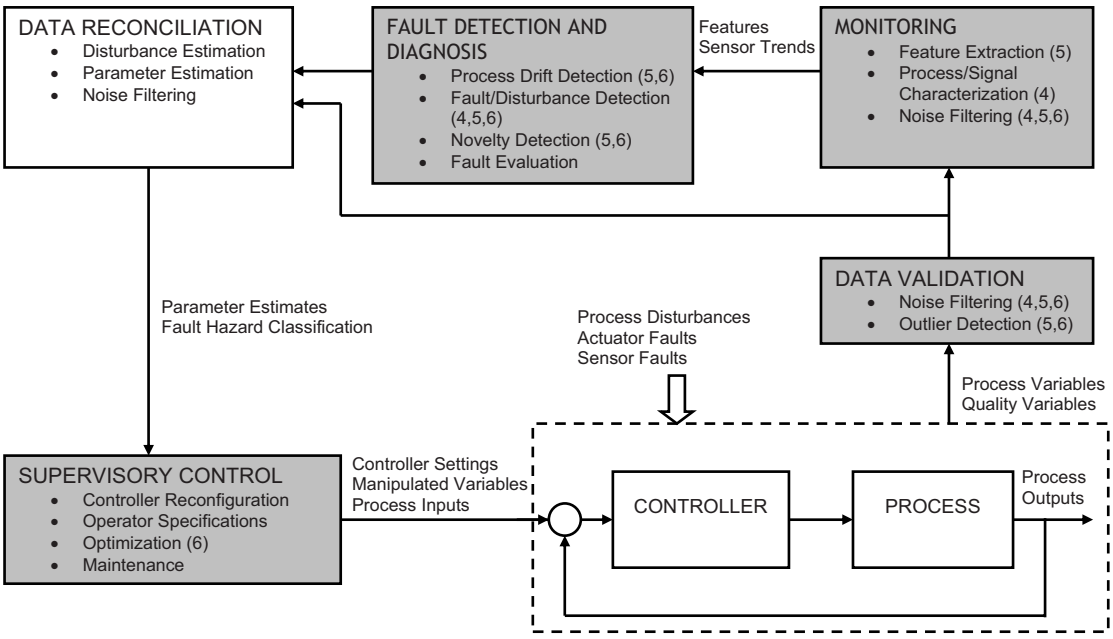

Figure 1.2: A high-level modular view of various areas of interest and their relationships within intelligent process control systems research and practice. The specific issues investigated in this thesis fall within the shaded blocks, the number(s) next to the tasks indicating the relevant corresponding chapter(s).

1.7 Organization of Thesis

In the next chapter, a literature review of progress and trends in monitoring, analysis, and fault diagnosis of process operations is given. The general formulation of the fault diagnosis is presented. The different modelling approaches that have been considered are discussed. Because of their strong connections to the present work, a relatively detailed exposition on the use of multivariate statistical methods and neural networks is given.

A comprehensive discussion of the theoretical and algorithmic aspects of learning from data paradigm is presented in Chapter 3. The concept of large margin optimization necessary

for improved generalization performance, as well as the flexibility introduced by kernels are discussed. Basic kernel-based algorithms for both supervised and unsupervised learning are introduced.

Application of the theoretical and algorithmic framework starts in Chapter 4 where an improved method for classification of time series using nonlinear singular spectrum analysis (SSA) is presented. Furthermore, the enhanced noise filtering properties of nonlinear SSA are highlighted. The usefulness of the method in system identification in process engineering is discussed by way of examples.

Projective latent methods are well-established in multivariate statistical process monitoring (MSPM). A comparative analysis of linear and nonlinear latent variable projective methods is presented in Chapter 5. Also, a delineation of normal operating regions in statistical process monitoring (SPM) charts based on estimating the support of a distribution is proposed. Fault diagnosis is analyzed from an unsupervised learning perspective using one-class classification methods. A framework for fault diagnosis using one-class SVMs in proposed, including a critical analysis of the approach using a representative model.

Chapter 6 discusses a method for continuous improvement of process operations using decision trees and support vector classifiers.

Finally, the thesis concludes highlighting main contributions of the study as well as recommendations for future investigations.

Chapter 2

Theory and Contemporary Practice in Diagnosis of Process Systems

Marquis Wu asked: "What measures will ensure the soldiers will be victorious?"
Wu Ch'i replied: "Control is foremost... If the laws and orders are not clear; rewards and punishments not trusted; when sounding the gongs will not cause them to halt or beating the drum to advance, then even if you had one million men, of what use would they be?"

Excerpt from Wu-Tzu, *translated by Ralph D. Sawyer*

Speaking as a control engineer, I... welcome this flirtation between control engineering and statistics. I doubt, however, whether they can yet be said to be 'going steady'.

J.H. Westcott (1962)

THE main objective of a fault diagnostic system is early detection, isolation and/or identification of process faults to avoid complete failure of a physical system and its subsystems. Failure to detect and correct faulty conditions has an adverse effect on the safety, reliability, efficiency and product quality of process operations. Stimulated mainly by progress in modern control theory as well as challenges arising from automation and complexity of modern-day plants, a unified methodology for analysis of the fault diagnosis problem is now in place. Central in the approach is an information processing module aimed at extracting fault information from available knowledge about the process, ideally summarized in the form of a mathematical model relating system inputs and parameters to measured outputs. Success of any diagnostic procedure is, therefore, interrelated to how well the model explains observed behavior of a given process under normal operating conditions.

Although the task of fault diagnosis has its foundations in control systems engineering, where it is also referred to as an FDI (fault detection and isolation) or FDIA (fault detection, isolation and analysis) system depending on its application (Frank, 1990; Frank et al., 2000), it has now assumed an inter-disciplinary character because of the need to exploit various kinds of knowledge found in different operating environments. More specifically, progress in computational learning theory, statistical pattern recognition and system identification has availed many modelling methods that use and capture different forms of knowledge. Therefore, for any given situation different diagnostic procedures based on different model representations can be constructed.

In this chapter the model-based fault diagnosis methodology is briefly reviewed. First, the analytic-based redundancy approach that forms the basis of the methodology is presented and its limitations highlighted. An overview of the knowledge-based and data-driven approaches then follows. Given their close relationship to the present study and widespread application in chemical and metallurgical processes, relatively detailed reviews on the use of artificial neural networks and multivariate statistical process monitoring in diagnosis of technical processes are given.

2.1 Basic Principles of Model-Based Fault Diagnosis

The goals of fault detection, identification and diagnosis are respectively early warning of occurrence of a fault or abnormal system behavior, identifying/isolating the variables(s) responsible for triggering the alarm, and finding the root cause(s) of the abnormal behavior. These three tasks are centered around a (dynamic) process plant that consists of actuators, sensors and plant dynamics (referred to as components), which given (known and unknown) inputs executes a chain of activities to yield certain outputs (see Figure 1.1). While process monitoring focuses on detection and identification, diagnosis provides the necessary interpretation of the fault information to generate guidelines for intervention or supervisory control. Fault diagnosis involves monitoring a system by modelling the following possible effects that may indicate abnormal plant behavior:

- faults in the actuators, sensors, or plant dynamics;
- errors induced by the mismatch between process and process model; and
- measurement and plant disturbances.

In earlier times, process systems had relatively simple configuration and many of the tasks were manually operated. Fault detection and diagnosis were restricted to ensuring directly measurable variables did not violate fixed limits or trends. With increasing automation more sophisticated diagnosis methods based on signal processing, hardware redundancy and plausibility schemes were introduced (Isermann, 1984). However, as processes have become very complex and highly automated these approaches are now inadequate because of costly implementations, limited operating ranges, and poor efficiency. Because of these limitations, model-based fault diagnosis techniques have been proposed and developed over the last few decades based on modern control theory (Frank et al., 2000). In its basic formulation, model-based fault detection and diagnosis involves two steps of residual generation and residual evaluation, Figure 2.1.

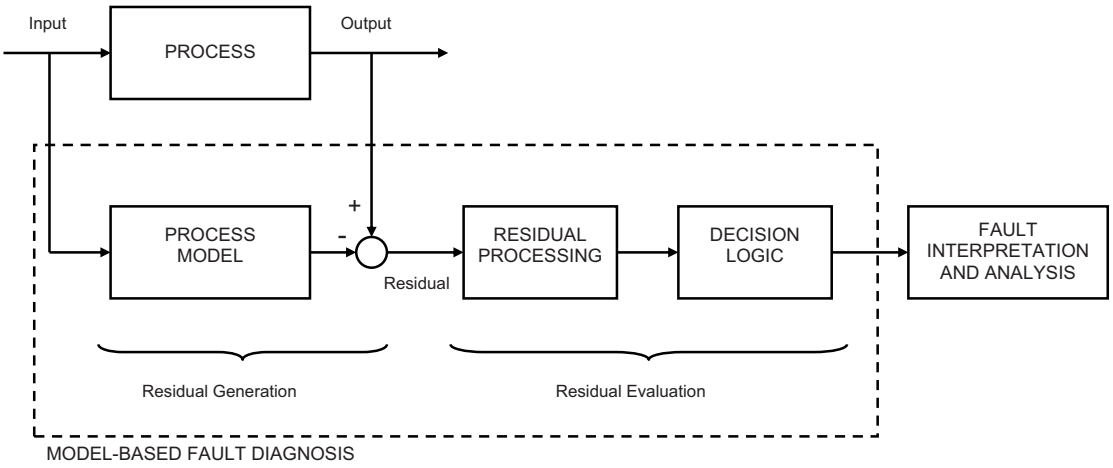

Figure 2.1: Functional view of model-based fault diagnosis.

2.1.1 Residual Generation

In model-based methods, a mathematical model is sought that accurately describes the nominal static and dynamic relationships between system inputs and measured outputs. The model is derived analytically using fundamental knowledge of the process under consideration, which can simply be considered as input-output relations of the dynamics of the system. During operation, the model is presented with the same inputs as the process to give an estimate of the observed variables, which are then compared to the actual measured variables to generate symptoms or residuals. The residuals reflect the impact of faults on the process.

Because an exact mathematical model is impossible to derive for any real process, and knowledge of all the inputs is rarely available, the residuals are mixed with other signals which do not contain information about faults. Therefore, designing model-based diagnostic systems requires a residual generator that is sensitive to faults of interest and simultaneously robust to the influence of model uncertainties and disturbances. Residual generators with high sensitivity to faults and robustness to unknown disturbances have been implemented using state estimation (i.e., parity check, observer schemes, detection filters) and parameter estimation techniques (Frank et al., 2000; Isermann and Ballé, 1997; Isermann, 2005; Patton, 1997).

2.1.2 Residual Evaluation

Subsequent to residual generation is the evaluation of residuals to decide the likelihood of whether a fault has occurred. If knowledge of all the inputs to the process is available and the process model is exact, then the following fault decision logic is sufficient to detect a fault:

if $\mathbf{r}(t) \neq 0$ **then**,
    fault has occurred
else

no fault,
end if

where $\mathbf{r}(t)$ is the residual vector at time t. Because of unavoidable modelling uncertainties as well as system/measurement noise, robust evaluation is required to avoid or minimize false alarms and failure to detect process deviations from normal conditions. The residual evaluation procedure involves choice of the evaluation function and corresponding detection thresholds. During process operation, the output from the preceding residual generation step is input into a residual evaluation function. The resulting output is compared with the threshold limits and, depending on whether or not a violation of the threshold has occurred, an appropriate decision is made. Thus the algorithm presented previously is modified to:

if $g(\mathbf{r}(t)) \geq \theta$ **then**
fault has occurred
else
no fault,
end if

where $g(\cdot)$ is the evaluation functional and θ a threshold. Different evaluation functionals have been considered in literature including simple threshold tests, moving averages of the residuals, norm-based methods and methods based on statistical decision theory, for example generalized likelihood ratio test or sequential probability testing (Basseville and Nikiforov, 1993; Frank et al., 2000; Patton et al., 2000).

Note that two fault modes (time evolution) can be distinguished: (a) abrupt or sudden faults, for example a blocked valve, and (b) incipient or slowly developing faults, for example process drift or sensor bias. Depending on the targeted application, the appropriateness and importance of the different modes may vary.

2.1.3 Nonlinear Model-Based Fault Diagnosis

Although real processes are typically nonlinear, there is no general theory yet for handling nonlinearity in fault detection and identification problems using mathematical models (Frank et al., 2000). To generate residuals for nonlinear processes, a widely practiced approach in control engineering is reducing the problem to a linear one using linearization techniques. Subsequently, robust and adaptive state estimation and parameter estimation techniques are then applied. Unfortunately, the extent of linearization of any system is limited. Moreover, linearization errors increase model uncertainties, resulting in poor performance and reliability of the fault diagnostic system. It has been pointed out that linearization tends to work for well-defined processes such as those found in aeronautical, mechanical, and electrical systems (Isermann, 1984; Venkatasubramanian, 2005). For highly nonlinear systems, for example chemical and metallurgical processes, the standard model-based scheme is problematic, and other approaches need to be considered.

An alternative is to replace the analytical model with models inspired by advances in information processing systems. In the next sections knowledge-based and data-driven methods are introduced and a few specific approaches reported in literature are discussed.

2.2 Knowledge-Based Redundancy

The main limitation of analytical approaches is the requirement of an exact mathematical process model, which is difficult to obtain in practice. Although robust and adaptive design techniques have been suggested, an alternative approach is that of using methods inspired by artificial intelligence (AI), particularly for complex processes with inadequate process knowledge. Knowledge-based systems (KBS) or expert systems were among the first group of AI technologies to be applied to plant control systems.

From a KBS perspective, fault diagnosis is basically a reasoning activity that involves the mapping of symptoms ("residuals") to a hypothesis space developed through experience with the system (Prasad and Davis, 1992). Thus, instead of functional reasoning as in model-based approaches, a rule-inferential system is implemented using expert knowledge in the form of heuristic rules and data stored in a knowledge base. Rules take the form of logical event chains describing cause-effect relationships. The performance of knowledge-based diagnostic systems is dependent on the accuracy and completeness of the knowledge base.

A number of methods implementing rule-based diagnostic systems have been developed for chemical process systems (Petti et al., 1990; Stephanopoulos and Han, 1996; Venkatasubramanian and Rich, 1988). However, applications have been limited to targeted processes. This can be attributed to a number of reasons including the difficult and costly exercise of knowledge acquisition from operators; lack of generality; inability of an expert system to adapt or dynamically improve its performance; and inability to handle novel situations (Joseph et al., 1992; Venkatasubramanian and Chan, 1989). Since the KBS has the appealing property of interpretable solutions, the current trend is to optimize the design of knowledge-based diagnostic systems by integrating it with other technologies such as analytical modelling, fuzzy logic, machine learning, and pattern recognition techniques (Frank, 1990; Frank et al., 2000; Özyurt and Kandek, 1996; Uraikul et al., 2006). Before discussing the hybrid approaches, data-driven approaches are presented first.

2.3 Data-driven Diagnostic Methods

2.3.1 Artificial Neural Networks

A Brief Introduction

Artificial neural networks (ANNs) are among the most widely used nonlinear learning algorithms inspired by Frank Rosenblatt's linear perceptron algorithm for classification (Rosenblatt, 1959). Although the linear perceptron was introduced more than 50 years ago, ANNs became popular only after Rumelhart et al. (1986) suggested a computationally feasible algorithm for solving linearly non-separable problems using a network of perceptrons called multilayer perceptron (MLP). It had earlier been argued that no computationally feasible algorithm could be realized for solving these problems (Minsky and Papert, 1969).

Structurally, an artificial neural network consists of many exhaustively connected simple processing units called neurons, each possibly having a limited memory capacity. Information processing is achieved by mapping a measurement space to the output or decision

space via the interconnections in the network. In the case of feed forward networks, the output of the network can be represented as explicit mathematical functions of the inputs and strengths between various connections, known as weights. Hence, feed forward networks represent generalized nonlinear functional mappings between input and output variables (Bishop, 1995).

The weights between the processing units are determined through a process called training using a finite set of patterns sampled from the generating but unknown source. During training the weight values are recursively adapted to "learn"[(1)] a rule that in the ideal case generates the correct output given an instance of the input data. Thus, the weights contain the knowledge of the underlying joint density function of the input and output data. When presented with a previously unseen input instance, the trained network attempts to generate the associated set of outputs. Because of this pattern recognition capability, it is not uncommon to see comparisons being drawn with biological neural networks.

In addition to learning, artificial neural networks possess other computational properties attributable to "intelligent" behavior such as association, generalization, detecting novel patterns, self-organization, pattern discrimination and self-stabilization (Joseph et al., 1992; Kohonen, 1995; Venkatasubramanian and Chan, 1989). Moreover, because of the distributed parallel information processing structure, ANNs are capable of solving complex problems rapidly. Figure 2.2 shows an example of a feed forward multilayer perceptron network.

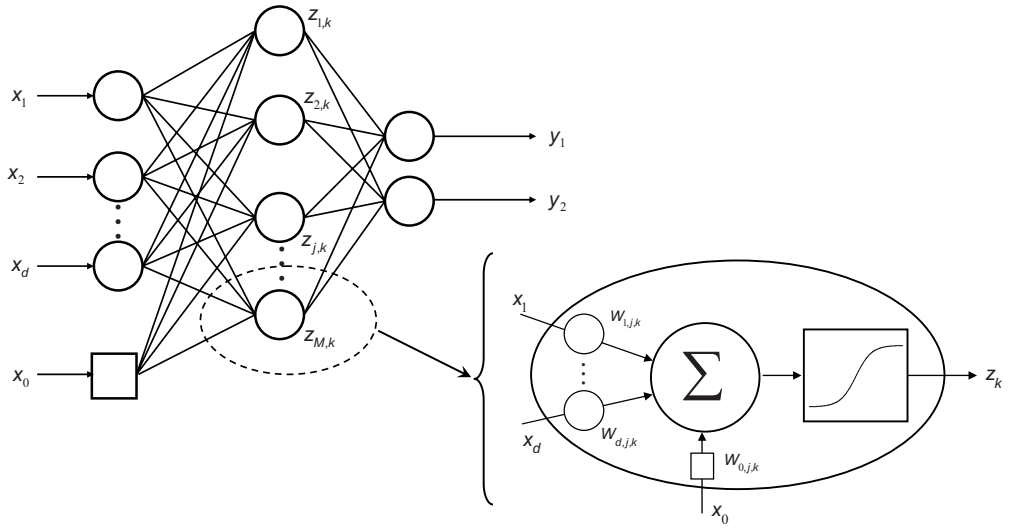

Figure 2.2: A typical feed forward multilayer network with one hidden layer and an output layer. The relationship between the input variables $\mathbf{x} \in \mathbb{R}^d$, the bias x_0 and output variables $\mathbf{y} \in \mathbb{R}^2$ is determined by minimizing a specified objective function and adaptively modifying the weights $W_{i,j,k}$ by repeated presentation of a training instance to the network until no changes are made to the weights. The output from each neuron z_k in the hidden layer is a weighted sum of each input processed processed by a suitable transfer function.

(1) A formal definition of what constitutes "learning" in a data-based modelling context can be found in Chapter 3.

Applications of Neural Networks in Fault Diagnosis

In principle, given enough representative data, artificial neural networks can approximate any mathematical function with very good accuracy. Because of this property, it has been proposed to use ANNs in fault diagnostic systems for nonlinear system identification and pattern recognition, that is residual generation and residual evaluation respectively (Sorsa and Koivo, 1991; Venkatasubramanian and Chan, 1989). A beneficial effect is that the requirement of an explicit mathematical model as required in parameter and state estimation methods is circumvented. As indicated before, chemical processes are highly nonlinear systems and deriving an exact mathematical model is very difficult because of incomplete knowledge.

Residual generation using neural models entails using an artificial neural network as a proxy for the analytical model. Thus, the model is identified using nominal input-output data for the system under consideration. Similarly, models are fitted for each known fault condition. In each case, the data is obtained by direct sampling from the process or, more likely, generated using a realistic simulation model of the process. The identified bank of neural network models are then deployed on the plant for residual generation. Fault diagnosis follows residual generation by solving a pattern classification problem. Figure 2.3 is a schematic illustration of the general fault diagnosis scheme using a bank of models.

An important point to consider in nonlinear system identification is incorporation of dynamics into the network structure. Although the problem of handling dynamics is still not yet completely resolved, two approaches are usually implemented in practice. Static ANNs can be endowed with memory by applying a bank of cascading filters to the network inputs. Since the resulting free parameters have fixed parameters, the learned model is strictly speaking a quasi-dynamic model. Alternatively, dynamic ANNs can be realized using neuron structures whose internal representation are adaptive and not fixed (Frank et al., 2000; Haykin, 1994). This is achieved through the use of time delay elements and recurrent connections. A framework for identification of nonlinear processes using dynamic neural networks was proposed in Shaw et al. (1997). Applications of recurrent dynamic networks in developing a fault diagnosis system for a sugar evaporation process have been reported (Patan and Parisini, 2005).

Gomm et al. (2000) evaluated the use of principal component analysis (PCA)[(2)] in preprocessing fault signals (or residuals) in an application of neural network-based fault diagnosis to an industrial nuclear fuel processing plant. The system involved is a complex multivariable process whose physical states and coefficients are mostly unknown and, therefore, an analytic model-based scheme could not be applied. PCA was used to reduce dimensionality of the network inputs and, as a result, obtain a parsimonious network topology. PCA filters random noise influences in data while reduced complexity of the network structure minimizes computational costs of training and storage. A potentially adverse issue not discussed in Gomm et al. (2000) is possible fault masking effect PCA may have on the resulting filtered data during testing. Since PCA discards information contained in the subspace explaining minimal variation in the data, incorrect diagnosis could result from use of filtered fault signals. This is particularly important for slowly developing faults that are

[(2)] See section 2.3.2

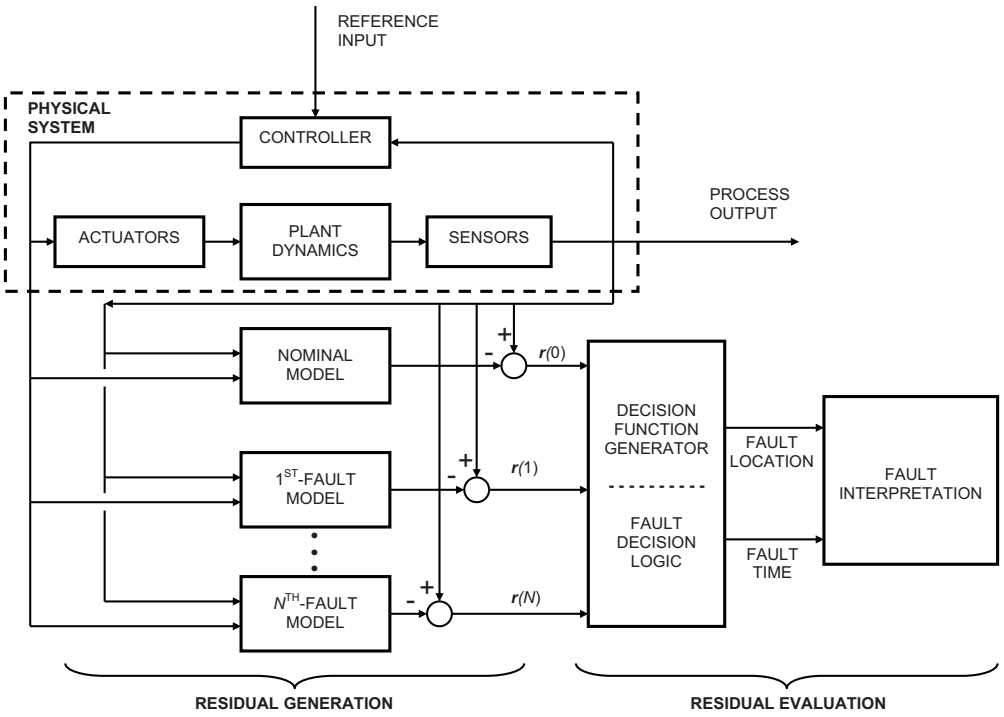

Figure 2.3: Fault diagnosis scheme using neural networks. A bank of models of the nominal (fault-free) and possible known faults conditions is constructed off-line and deployed for online residual generation and residual evaluation. The fault decision logic can also use a neural network model, the difference being that instead of nonlinear system identification, a pattern recognition learning problem is solved.

generally difficult to detect.

Venkatasubramanian and Chan (1989) applied supervised neural networks for fault diagnosis in a pattern recognition context. Using knowledge of different faults from a catalytic cracking unit used in petrochemical processes, it was shown that a trained network was able to diagnose correctly future faulty conditions. Moreover, multiple fault diagnosis was also possible even if the network had been optimized using knowledge of single fault conditions only. Generalization to multiple faults simplifies the training phase of the network, as also demonstrated in Watanabe et al. (1994) where a hierarchical ANN is used to simplify the training. However, the proposed methodology could not handle novel faulty conditions not previously seen during training. In a sequel, Venkatasubramanian et al. (1990) investigated the robustness and fault tolerance capabilities of multilayer perceptrons in the presence of noise and sensor malfunction. In both cases, only steady-state processes were considered, and the direction and magnitude of changes in the measurements were not taken into account.

An investigation of the performance of different neural network architectures for fault diagnosis indicated that the multilayer perceptron provided the most reliable architecture (Sorsa and Koivo, 1991). In further investigations, Sorsa and Koivo (1993) did a comparison of supervised and unsupervised neural nets, for example Kohonen's self-organizing map

(SOM) (Kohonen, 1995) and adaptive resonance theory (ART) architectures (Carpenter and Grossberg, 1990). It was shown that supervised networks achieved better classification. Unsupervised neural network models were suggested for use in process classification since all possible faults could not be known a priori. Jämsä-Jounela et al. (2003) developed a fault diagnosis system by combining a knowledge-based scheme with self-organizing maps that was subsequently applied to a copper flash smelting process. Process knowledge obtained from plant operators' experiences was used to categorize neurons in the SOM.

Limitations of Neural Networks

The objective function used for determining optimal weights for neural network models is non-convex. Therefore, the training process often results in suboptimal models due to entrapment in local minima (Bishop, 1995; Haykin, 1994). In practice, a number of heuristic techniques such as early stopping or cross-validation, are used to diminish the local minima artefact.

Neural networks are sometimes lumped in the group of so-called "black-box" modelling techniques as they do not give insight into fundamental understanding and knowledge of a process. In particular, it is not possible to relate network topology or magnitudes of the weights to some physical aspect of the process. The lack of easily interpretable solutions of neural network models limits their adoption by operators, especially for purposes of supervisory control.

Kramer (1992) studied the performance of feed forward neural networks under conditions typically encountered in industrial practice, for example corrupted and limited training data sets. The following structural weaknesses of ANNs relevant to the fault diagnosis problem were identified.

- The decision function obtained after training was influenced by data points located on the boundary resulting in linear decision functions for small training sets even when the true underlying function was nonlinear.
- Poor generalization of the networks was observed that was attributed to the tendency of ANNs to arbitrarily place the decision boundary in empty regions of input space, leading to large extrapolation errors. In addition, the decision boundaries for the normal class remained unbounded. In a later investigation, Rengaswamy et al. (2001) showed that the effect of the unbounded normal class problem could be reduced by use of ellipsoidal activation functions, which are based on a distance metric and, therefore, more robust.
- By inducing different kinds of perturbations in the fault class distributions, the trained model exhibited higher sensitivity when compared to distance-based classifiers.

2.3.2 Multivariate Statistical Process Monitoring

Statistical-based models represent another group of data-driven techniques that are widely used in stochastic processes such as parts manufacturing as well as multivariate systems. Statistical process control (SPC) is a well-established methodology used for process improvement by detecting and eliminating root causes of variability associated with a process

(Tucker et al., 1993; Vander Wiel et al., 1992). This is accomplished by monitoring key product or process quality variables with the aid of statistical monitoring charts, for example Shewhart, exponentially weighted moving average (EWMA) and cumulative sum (CUSUM), that can distinguish between *common cause* variation and *special* or *assignable causes* (Box and Kramer, 1992). It is usually assumed that the quality measurements are independent and identically distributed (i.i.d.) and the goal is to monitor unusual variations from the model. Therefore, SPC is a type of hypothesis testing in which common cause variation is considered consistent with an *a priori* specified null model corresponding to a "state of statistical control." On the other hand, special events indicate significant process deviations from the model. In such a case, operator intervention is required to search and investigate the cause(s) of such a special event(s). SPC must be contrasted to regulatory control which is a process optimization technique where the objective is to maintain set points of important parameters through compensatory adjustment using feedback controllers.

While classical SPC considers only quality data, because of progress in instrumentation and computer technologies it is not atypical for modern-day plants to routinely measure and collect data on many variables, in some cases in orders of magnitude of a thousand records per second (Venkatasubramanian, 2005). This is particularly true for process variables that are measured continuously throughout the process such as temperatures, flow rates and pressures. These variables contain important information on the propagation of phenomena within the process and need to be considered in plant monitoring strategies. Unfortunately, the multidimensional nature of these variables makes them less suitable for analysis in the classical SPC framework. Moreover, these variables tend to be highly correlated since they all result from similar underlying driving forces. This lack of independence confounds analysis and interpretation of their respective statistical monitoring charts. Although multivariate extensions of Shewhart, CUSUM, and EWMA charts have been proposed, these do not work with data containing redundant information (Ku et al., 1995; MacGregor and Kourti, 1995). Hence, the obtained charts are likely to be misleading on the true state of the process. To handle multivariable continuous processes, the use of multivariate statistical process control (MSPC) methods in process systems has received much attention in the last 10–15 years (Kourti and MacGregor, 1995; Kresta et al., 1991; MacGregor and Kourti, 1995; Wise and Gallagher, 1996).

Multivariate statistical control methods are based on the statistical projection methods of principal component analysis (PCA) and partial least squares (PLS) (also referred to as projection to latent structures in some contexts). PCA and PLS handle large numbers of highly correlated variables, possibly corrupted with measurement noise, by finding a low dimensional subspace explaining the dominant variability in the data. The orthogonal residual subspace is then considered to be due to high frequency components in the data. In addition to extracting the most descriptive features of variation in data, PCA has also been used to estimate missing values in data (Jolliffe, 2002). It is the most common method used in MSPC methods for process monitoring, control, and diagnosis (Kresta et al., 1991; Wise and Gallagher, 1996). In the next section a derivation and mathematical properties of PCA are presented. Extensions to the closely related technique of PLS have also been formulated but are not discussed here.

Principal Component Analysis

Principal component analysis (PCA) is a well-established multivariate technique used for feature extraction and dimensionality reduction from multi-dimensional data (Fukunaga, 1990; Jolliffe, 2002). It is based on the eigen-decomposition of the sample covariance matrix of a data matrix $\mathbf{X} \in \mathbb{R}^{m \times d}$ given by

$$\lambda_i \mathbf{p}_i = \mathbf{C}\mathbf{p}_i, \qquad i = 1, \ldots, d \tag{2.1}$$

where $(\lambda_i, \mathbf{p}_i)$ is the i^{th} eigenvalue-eigenvector pair, arranged in non-increasing order of the eigenvalues. The covariance matrix $\mathbf{C}$ is defined as

$$\mathbf{C} = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})'. \tag{2.2}$$

where $\bar{\mathbf{x}}$ is the mean vector of the variables and m the number of observations.

Because of cross-correlations and low signal-to-noise ratios, it is often that only a few dominant principal directions explain maximum variation in the data. Hence, a compact representation can be obtained by retaining only the dominant eigenvectors. Supposing that q directions explain maximal information based on the eigenvalues, the original matrix can then be decomposed according to

$$\mathbf{X} = \sum_{j=1}^{q} \mathbf{t}_j \mathbf{p}_j{}' + \sum_{j=q+1}^{d} \mathbf{t}_j \mathbf{p}_j{}' \tag{2.3}$$

$$= \mathbf{T}\mathbf{P}' + \tilde{\mathbf{T}}\tilde{\mathbf{P}}' \tag{2.4}$$

$$= \mathbf{T}\mathbf{P}' + \mathbf{E} \tag{2.5}$$

where $\mathbf{t}_j$ and $\mathbf{p}_j$ are vectors of the principal components or scores and loadings respectively, $(\mathbf{T}, \tilde{\mathbf{T}})$ the matrices of scores, $(\mathbf{P}, \tilde{\mathbf{P}})$ the loadings matrices or principal directions, and $\mathbf{E} = \tilde{\mathbf{T}}\tilde{\mathbf{P}}'$ is the residual matrix after projecting the data onto the principal component subspace defined by the leading q principal directions.

The subspace identified by PCA possesses some interesting mathematical and statistical properties (Burges, 2005; Jolliffe, 2002). First, scores or projections onto the leading q eigenvectors explain maximal variance than all other q orthogonal directions. Also, the PC subspace is optimal in that the mean-squared approximation error ($\|\mathbf{E}^2\|$) in representing the observations by the first q principal components is minimal over all possible q directions. Furthermore, the principal components are uncorrelated, which has important consequences in the extension of SPC techniques to the multivariate case. More specifically, each score variable has the simplicity of representation and interpretation as in classical univariate statistical monitoring charts. Finally, PCA maximizes mutual information for Gaussian distributed data, which makes it a useful pre-processing technique in, for example, blind-source separation applications.

There is no universally accepted approach to the question of selecting the optimal number of principal components q. A number of approaches have been proposed including eigenvalue

thresholding, scree plots, parallel analysis, and cross validation (Ku et al., 1995). Raich and Çinar (1996) used an F-test method that looks for an elliptical rather than spherical shape for the confidence bound to determine the number of dimensions. Recently, Minka (2001) proposed a method based on re-interpreting PCA as a density estimation problem and solving a Bayesian model selection problem to estimate the true dimensionality of the data.

Process Monitoring Based on PCA

Given a reference data set $\mathbf{X}$ taken from a process when the operating conditions were under a "state-of-statistical control", the MSPC approach fits a PCA model to the data to define a normal operating region for the process. Multivariate statistical process monitoring charts can be developed for the scores, sum of scores, and residuals by defining corresponding statistical control limits. The multivariate Hotelling's T^2 statistic characterizes the deviation of a process from the expected behavior of the process under normal operating conditions and is given by

$$T^2 = \sum_{i=1}^{q} \frac{t_i^2}{\lambda_i}. \tag{2.6}$$

where t_i is the score corresponding to the i^{th} eigenvector and λ_i the associated eigenvalue. An out-of-control situation is indicated if the the T^2-statistic in Equation (2.6) exceeds the following control limit

$$T_\alpha^2 = \frac{p(m-1)}{m-p} F_{p,m-1,\alpha} \tag{2.7}$$

where $F_{p,m-1,\alpha}$ is the upper $100 \cdot \alpha\%$ critical point of the F-distribution with p and $n-p$ degrees of freedom (MacGregor and Kourti, 1995; Wise and Gallagher, 1996).

Hotelling's T^2 statistic is useful for quantifying the variation of an observed sample within the principal component subspace. It is also possible to derive a statistic for the expected distribution of projections in the residual subspace of an "in-control" sample (Jackson and Mudholkar, 1979). The residual vector between a sample and its principal components is

$$\mathbf{e} = (\mathbf{x} - \mathbf{P}\mathbf{P}'\mathbf{x}) \tag{2.8}$$

with the sum of squares of the residuals obtained as

$$\begin{aligned} Q &= \mathbf{e}'\mathbf{e} \\ &= \mathbf{x}'(\mathbf{I} - \mathbf{P}\mathbf{P}')\mathbf{x} \end{aligned} \tag{2.9}$$

It can be shown that the quantity

$$c = \frac{\theta_1[(Q/\theta_1)h_0 - 1 - \theta_2 h_0(h_0-1)/\theta_1^2]}{\sqrt{2 \cdot \theta_2 h_0^2}} \tag{2.10}$$

is approximately normally distributed with unit variance and zero mean (Jackson and Mudholkar, 1979; Wise and Gallagher, 1996). Hence, the control limit for the sum of residuals

statistic or Q statistic is obtained as

$$Q_\alpha = \theta_1 \left[\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0} \tag{2.11}$$

where c_α is the normal deviate corresponding to the upper $(1-\alpha)$ percentile, $\theta_i = \sum_{j=p+1}^{m} \lambda_j^i$, for $i = 1, 2, 3$ and $h_0 = 1 - (2\theta_1\theta_3)/3\theta_2^2$.

The squared prediction error (SPE) or Q statistic in Equation (2.11) gives the distance from the principal component subspace and is a measure of variation in the residual subspace, which is complement to Hotelling's T^2 statistic. Figure 2.4 is an illustration of these statistical measures for a two-dimensional principal subspace obtained from data in $\mathbb{R}^3$.

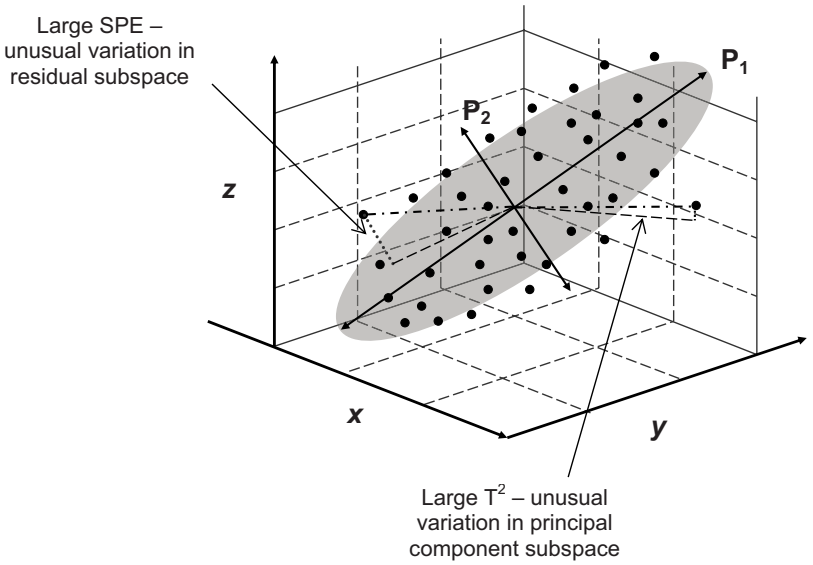

Figure 2.4: A 3D illustration of PCA and associated statistical monitoring quantities of T^2 and Q statistics. The shaded ellipse represents the 2D principal component subspace.

Advances in Multivariate SPC

Since the introduction of multivariate statistical approaches to monitoring and diagnosis of process systems in the early 1990s, there has been many innovative developments in the practical application of MSPC. This has largely been as a result of its rapid acceptance and use by the industrial community (Kourti et al., 1996). In the context of the fault diagnosis framework presented at the beginning of the chapter (section 2.1), these improvements to the basic MSPC framework cover both the residual generation as well as residual evaluation phases. In the following paragraphs, a brief outline of some of the developments are presented.

MacGregor et al. (1994) proposed a multiblock approach to handling large sets of variables whereby process variables are partitioned into different blocks corresponding to different process units or sections. Multivariate monitoring charts for individual units or subsections

of the plant, as well as for the entire plant can then be developed. By using multi-way projection method (Bro, 1997), MSPC techniques have been extended to batch processes. MSPC methods have also been developed for multistage processes by combining multi-way and multiblock approaches (Kourti and MacGregor, 1995).

Ideally, fault detection should trigger an automated search process for the special event and recommend corrective action. Unfortunately, the basic frameworks of both univariate and multivariate statistical process monitoring do not provide a fault identification or isolation step subsequent to fault detection. For process improvement, Dunia and Qin (1998) pointed out that both fault identification and diagnosis are more critical than detection. MacGregor et al. (1994) proposed a diagnostic method that is based on variable contributions to the SPE. In the analysis, variables with large absolute magnitude of the SPE are considered potential sources of observed faults. Since contribution plots are based on a non-causal correlation model, they do not provide for direct fault identification (Yoon and MacGregor, 2000), but rather narrow the search space of possible faulty variables.

A different approach to both fault detection and identification was proposed in Dunia et al. (1996). Instead of the the T^2 statistic, an EWMA for the SPE was used for fault detection that facilitated development of a validity index for sensor fault identification. The method involves generating a model for each possible sensor fault, reconstruction of each sensor in the event of a fault, residual examination and, finally, identification using the validity index. The residual generation uses a bank of possible fault models and the nominal model, in similar spirit to Figure 2.3. In a later investigation, explicit conditions necessary for fault detectability, fault reconstruction and fault identifiability were developed (Dunia and Qin, 1998).

Aldrich et al. (2004) and Gardner et al. (2005) proposed a related statistical process monitoring approach that emphasizes the visualization of process correlations and variations in process variables using the biplot methodology (Gower and Hand, 1996). The biplot is a multivariate analogue of the scatter plot. In addition, the approach provides for automatic detection and visualization of process disturbances by use of bagplots (Rousseeuw et al., 1999).

Principal component analysis is a powerful technique for decorrelating multidimensional data. Invariably, process variables are not only cross-correlated among each other, they also exhibit autocorrelation. This may arise from, among other, the high frequency of sampling, random noise, effects of feedback control, and other unknown plant disturbances. Applying PCA to such data does not remove the autocorrelation and, therefore, the i.i.d. assumption is violated resulting in high rates of false alarms as well as misses. Kresta et al. (1991) and Ku et al. (1995) proposed use of a lagged variable data matrix to eliminate autocorrelation. Negiz and Çinar (1997) used stochastic realization and canonical variate analysis to develop a statistical process monitoring method suitable for large data sets exhibiting autocorrelation and cross-correlation. The advantages of the method were illustrated by application to a milk pasteurization process.

A closely related concept to autocorrelation is the multiscale nature of data. While conventional PCA assumes a single time-frequency localization at all locations (or scale), chemical processes are multiscale in nature. Single scale representation has an adverse effect on the performance of PCA since an embedded error proportional to the number of retained

components always affects PCA (Tipping and Bishop, 1997). Bakshi (1998) introduced multiscale PCA for multivariate statistical monitoring that integrates PCA and wavelet analysis (Mallat, 1989). In particular, multiscale PCA has both a decorrelation effect due to PCA as well as deterministic feature extraction and de-autocorrelation capabilities of wavelet analysis. Yoon and MacGregor (2004) extended multiscale MSPC to identification of faults. A method for fault detection based on multiscale analysis and clustering-based diagnosis was introduced by Aradhye et al. (2002). A theoretical analysis of multiscale SPC based on wavelet analysis can be found in Aradhye et al. (2003).

Conventional PCA is a linear technique that extracts linear correlations in multidimensional data. In cases where the data exhibits nonlinear correlations successful application of MSPC may be restricted because of the inadequacy of linear PCA in explaining the nonlinear structure. Kramer (1992) proposed an auto-associative neural network for extracting nonlinear features that essentially consists of two serially arranged feed forward multilayer perceptrons whose input and output are similar. In a later study, Dong and McAvoy (1992) proposed a nonlinear PCA approach that integrates the principal curve algorithm (Hastie and Stuetzle, 1989) and neural networks. The main contribution of their approach was a method for generating an explicit nonlinear PCA loadings representation for the principal curve algorithm. Jia et al. (1998) used an input-output neural network for nonlinear PCA. Recently, Cho et al. (2005) and Choi et al. (2005) extended the use of the kernel-based nonlinear PCA algorithm (Schölkopf et al., 1998) to fault detection and diagnosis of industrial processes.

With respect to residual evaluation, the confidence limits for scores, sums of scores, and residuals are based on the very restrictive assumptions of normality and independence. Hence, it is often the case that the 'in-control' region is very conservative resulting in many potential alarms going unnoticed. Martin et al. (1996) and Chen et al. (2000) proposed estimating the control limits by bounding the "in-control" region nonlinearly using kernel density estimation and Monte Carlo sampling techniques.

2.3.3 Process Diagnosis Using Fisher Discriminant Analysis

Fisher linear discriminant analysis

Discriminant analysis is a statistical technique for finding a compact set of latent variables or features that best discriminate groups of data. It was originally introduced by Fisher (1936) and, hence sometimes referred to as Fisher discriminant analysis (FDA). Formally, the method involves maximizing the ratio of between-class variance to within-class variance of labeled objects. Given the training data

$$\mathcal{T} = [(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)], \text{ where } (\mathbf{x}, y) \in \mathbb{R}^d \otimes \mathcal{G} \tag{2.12}$$

where $\mathcal{G}$ is the set of possible labellings, discriminant analysis seeks a transformation $\mathbf{w}$: $\mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ for $d' \leq d$, where d is input space dimensionality, such that the groups in the data are optimally separated. This transformation can be expressed mathematically as (Fukunaga, 1990)

$$f(\mathbf{x}) = \mathbf{w}'\mathbf{x} + b \tag{2.13}$$

where $\mathbf{w}$ is the projection or weight vector and b a bias term. The weight matrix is chosen to maximize the cost function known as Rayleigh's coefficient

$$\vartheta(\mathbf{w}) = \frac{\mathbf{w}'\mathbf{C}_B\mathbf{w}}{\mathbf{w}'\mathbf{C}_W\mathbf{w}} \tag{2.14}$$

where $\mathbf{C}_B = \sum_{i \neq j}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)'$ and $\mathbf{C}_W = \sum_{i \in \mathcal{G}_i}\sum_{x \in S}(\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)'$ are respectively the between-class and within-class scatter matrices, $\boldsymbol{\mu}_i = m_i{}^{-1}\sum_{k \in \mathcal{G}_i}\mathbf{x}_k$ is the i^{th} group's mean vector, and m_i is the number of samples in group $\mathcal{G}_i$. Within-class scatter is the expected covariance of each of the classes while between-class scatter measures the expected covariance between the classes. The optimal $\mathbf{w}$ gives the direction that maximizes the distance between the projected class centers, while ensuring the within-class scatter in this direction is as small as possible. Figure 2.5 illustrates the basic idea in FDA.

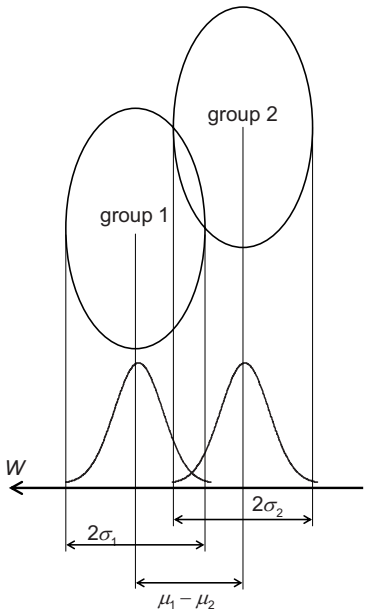

Figure 2.5: Discriminant analysis illustration for a 2-class problem. The objective of FDA is finding a direction $\mathbf{w}$ that maximizes separability of the different classes and simultaneously minimizing the spread of each class after transformation.

The solution to Equation (2.14) can be found by solving a generalized eigenvalue problem, guaranteeing a globally optimal solution. If the classes are normally distributed with equal covariances the FDA solution is equivalent to Bayes' optimal solution (Fukunaga, 1990).

Application of FDA in Fault Diagnosis

Although FDA is a widely used statistical pattern recognition method, applications in process monitoring and diagnosis have rather been limited. FDA seeks directions that maximize separability of classes. Therefore, it can be used advantageously for fault visualization and, in particular, fault diagnosis which involves supervised classification in selecting the most probable fault among a class of different possible fault conditions (Chiang et al., 2000).

A related technique, biplot canonical variate analysis (CVA), has also been proposed for visualizing different process conditions (Aldrich et al., 2004; Gardner et al., 2005). It can be shown that CVA and FDA essentially solve a similar problem (Kuss, 2002). Another integrative approach used genetic algorithms with discriminant analysis for key variable identification (Chiang and Pell, 2004). The method was extended to cope with nonlinearity in process data by incorporating support vector machines (Chiang et al., 2004). Recently, Peter He et al. (2005) developed a fault diagnosis approach based on fault directions by use of pairwise FDA.

2.4 Integrated Fault Diagnosis Approaches

Each of the fault diagnosis approaches of model-based, knowledge-based and data-driven techniques are capable of explaining different facets of knowledge about a system. To accentuate the strengths of each method while suppressing the respective limitations, it is logical to develop an integrated framework that combines the various methods. This has long been recognized as unavoidable for robust diagnostic systems. Hence, in practice fault diagnostic systems are designed targeted for multiplicity and redundancy (Isermann, 1984; Stephanopoulos and Han, 1996). A conceptual framework for integrating the different technologies was proposed in Prasad and Davis (1992), where modularization was defined in terms of information processing tasks, each with its own distinct form of knowledge organization and problem-solving methodology.

A very successful integrative method has been that of combining neural networks and fuzzy systems (Patton et al., 2000). Hybrid neuro-fuzzy or fuzzy neural networks possess the desirable learning, adaptation, and approximation properties of neural networks as well as transparent representation of knowledge in the form of rules and approximate reasoning capabilities of fuzzy systems (Frank et al., 2000). Both symbolic and numeric knowledge are therefore taken into account in the integrated system. Applications of neuro-fuzzy systems in fault diagnosis of real processes have been reported (Ayoubi and Isermann, 1997; Özyurt and Kandek, 1996; Pfeufer and Ayoubi, 1997).

Traditional statistical methods like PCA have mostly been used in pre-processing data before fitting or learning a decision function. The use of multivariate statistical methods is now widely acknowledged and applied. The integration of MSPC with other methods such as neural networks and wavelets has been discussed above. Norvilas et al. (2000) developed an integrated process monitoring and fault diagnosis scheme that combines CVA statistical process monitoring method (Negiz and Çinar, 1997) with knowledge-based systems. A comprehensive comparative analysis between multivariate statistical process monitoring and model-based methods has been presented in Yoon and MacGregor (2000).

2.5 Concluding Remarks

Advanced statistical and machine learning approaches have received considerable attention in the detection and diagnosis of anomalous process behavior. This is largely due to lack of adequate fundamental knowledge as well as the highly nonlinear nature of most processes, particularly those encountered in chemical and metallurgical industries. In this

chapter, a review of progress and practice in monitoring and diagnosis of processes was presented. Different modelling methods exploiting fundamental knowledge, operator experience and knowledge, and redundancy in process measurements were discussed with particular emphasis on statistical methods and neural networks. Though the diversity of the available methods can be overwhelming for the practitioner, the different approaches can be formulated in a unified theoretical fault diagnosis framework. The diagnosis problem is then viewed as a two-step model-based task consisting of residual generation and residual evaluation, facilitating integration as well interpretation of the tools.

In the next chapter a statistical learning perspective forming the conceptual basis of the results presented in the thesis is presented. Also, algorithmic formulations of a few central methods are presented.

Chapter 3

Learning from Data: Foundations and Algorithms

There is nothing so practical as a good theory.

Kurt Zadek Lewin (1890-1947)

COMPUTATIONAL learning has proved a very fertile field in developing data analysis methods, both in terms of theoretical and practical advances. Parallel developments in statistical learning theory, regularization theory, and functional approximation analysis have seen the emergence of a principled theoretical basis for analysis of the learning problem. Kernel methods are a recent contribution to machine learning that are a direct result of, in particular, statistical learning theory insights. The support vector machine (SVM) is a typical example of a kernel method that has been applied with great success in diverse fields such as object recognition, bioinformatics, text categorization, and machine vision.

In this chapter, the theoretical foundation that motivated development of support vector learning is discussed. The key ideas of large margin learning bias and implicit evaluation of similarities in high dimensional feature space using kernels are presented. The standard support vector algorithm and its nonlinear extension are discussed. Also, extensions of the basic SVM learning framework to other supervised and unsupervised learning methods are outlined.

3.1 Learning Theory

3.1.1 Learning from Data: A Statistical Perspective

The learning problem can be posed in a very general sense as follows. Given an object or *learner* that is exposed to stimuli or events in some environment, the learner attempts to discover a general rule associating these stimuli to the consequent changes in its behavior. Such a general rule can be expressed as an abstract representation of the interaction between the object and the environment. The discovered "rule", however, may not be an

accurate representation of the environment–object interaction since it is based only on observations and probably other prior assumptions about the environment. Nevertheless, the rule is useful for many purposes such as predicting course of action when exposed to stimuli similar to previous experiences. This abstract notion of learning illustrates an often encountered situation in process engineering. This is especially true for complex systems where the fundamental laws governing the systems' behavior are unknown, and instead all one has are observations from the system. It is therefore important to have a general theory that formalizes learning.

The problem of inferring functional dependencies in observed data – or learning from data – by computational means can be traced back to the realization that the digital computer can be used as a system for manipulating symbols and modelling cognition processes of the brain (Dreyfus and Dreyfus, 1988). Turing (1950) was among the first to lay a framework for building "machines that think" or, in some loose sense, intelligent machines. It turns out, however, that Turing's specifications are rather too strong in practice. Instead of "machines that think" efforts have mostly been directed at "machines that learn". A very early example of such a machine is the linear perceptron (Rosenblatt, 1959). The introduction of the perceptron initiated theoretic analysis of the learning process, starting with the work of Novikoff (1962).

An important issue in learning is the general conditions under which a learning algorithm is guaranteed to perform as well for as-yet-unseen (testing) instances as for the seen or training instances. In a series of investigations, Vapnik and Chervonenkis characterized the conditions necessary for learning functions to generalize to unseen examples on the basis of minimizing the training error (Vapnik and Chervonenkis, 1964, 1968, 1971, 1974, 1981, 1991). Related results were also obtained in regularization theory for the solution of ill-posed inverse problems (Tikhonov and Arsenin, 1977).

The analysis of the learning problem in a statistical framework assumes that one has some sample data (Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002; Vapnik, 1998)

$$\mathcal{T} := \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\} \in \mathcal{X} \times \mathcal{Y} \tag{3.1}$$

of sample size m, generated independently from a fixed but unknown distribution $\mathcal{P}(\mathbf{x}, y)$ over the input space $\mathcal{X}$ and output space $\mathcal{Y}$. The only knowledge available about $\mathcal{P}(\mathbf{x}, y)$ is contained in the m samples. To estimate the conditional distribution function $\mathcal{P}(y|\mathbf{x})$, a learning algorithm chooses a deterministic hypothesis $f: \mathcal{X} \to \mathcal{Y}$ from an *a priori* specified set of functions or hypothesis space $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ on the basis of the training data. Here, $\mathcal{Y}^{\mathcal{X}}$ is the set of all possible mappings from $\mathcal{X}$ to $\mathcal{Y}$. The performance of the learning machine is assessed using an appropriately defined non-negative loss function $\ell(y, f(\mathbf{x}))$. The learning problem is thus formulated as finding the function f^* which minimizes the expected loss or risk functional

$$\begin{aligned} \mathcal{R}(f^*) &= \mathbf{E}\{\ell(y, f(\mathbf{x}))\} \\ &= \int \ell(y, f(\mathbf{x}))\, d\mathcal{P}(\mathbf{x}, y). \end{aligned} \tag{3.2}$$

By defining the allowable values for the targets (y) and an appropriate loss function $\ell(\cdot)$, the learning problem as formulated generalizes to include classical problems such as pat-

tern recognition or classification, function estimation or regression, and density estimation (Vapnik, 1998, 1999).

3.1.2 Empirical Risk Minimization and VC Theory

The learning problem as above cannot be solved directly because $\mathcal{P}(\mathcal{X}, \mathcal{Y})$ is unknown. Hence, an approximate solution can only be obtained on the basis of the available data as well as the properties of the hypothesis space $\mathcal{H}$. In particular, one wants to find a function f from the set of functions in $\mathcal{H}$ whose expected loss converges to the minimal actual risk (Equation 3.2) over all $f \in \mathcal{H}$ in the limit of infinite data ($m \to \infty$). An *induction principle* is required to choose such an f. The empirical risk minimization (ERM) principle is typically used to select an optimal f^* (in the sense described above) that minimizes the empirical error

$$\mathcal{R}_{\text{emp}}(f) = \frac{1}{m} \sum_{i=1}^{m} \ell(f(\mathbf{x}_i), y_i). \tag{3.3}$$

The ERM principle has been the main focal point of much research since the introduction of Rosenblatt's perceptron. Specifically, it was then argued that learning corresponded to choosing a network structure (coefficients or weights) with best performance on a training set, and achieving an optimal error on the training set automatically guaranteed similar performance on test data or generalization[(1)] (Vapnik, 2000). Unfortunately, a function with minimal empirical error (Equation 3.3) does not necessarily give the minimal actual risk (Equation 3.2) because of the over–fitting phenomenon (referred to as the bias-variance trade off or capacity control in some contexts). Briefly, given a large class of functions $\mathcal{H}$ which contains all possible mappings $f: \mathcal{X} \to \mathcal{Y}$, the optimal function chosen according to the ERM will retain zero training error. However, the test error may not converge to the training error if the function takes arbitrary values on test data. This is the key (though trivial) insight captured in the so-called "No Free Lunch Theorems"[(2)] (Wolpert and Macready, 1997). Without restricting the capacity of the hypothesis space, it is impossible to estimate the true underlying function using empirical data, as illustrated in Figure 3.1.

Statistical learning theory, which was developed mainly by Vapnik and Chervonenkis (see, e.g. Vapnik (1998, 2000)), provides a complete characterization of the necessary and sufficient conditions for the generalization and consistency of the ERM principle. Additionally, it provides for understanding and controlling the rate of convergence of $\mathcal{R}_{\text{emp}}(f)$ to the actual risk $\mathcal{R}(f)$. Using such a framework, it is possible to construct learning algorithms with improved generalization performance that optimize these quantities using finite data.

An important consideration in learning is *consistency*, or how well a learned model approximates the true underlying function as more training data becomes available. The following

(1) More formally, the i.i.d. assumption implies that a correlation between the performance on the training and testing data sets can be related by probability theory. In particular, confidence intervals for risk functional $\mathcal{R}(f)$ can be obtained on the basis of the corresponding $\mathcal{R}_{\text{emp}}$ for each $f \in \mathcal{H}$.

(2) In simple terms, given any two functions $f(a)$ and $f(b)$, there are "as many" targets (or priors over targets) for which $f(a)$ has lower expected training error than $f(b)$ and vice-versa, for loss functions like 0/1 loss.

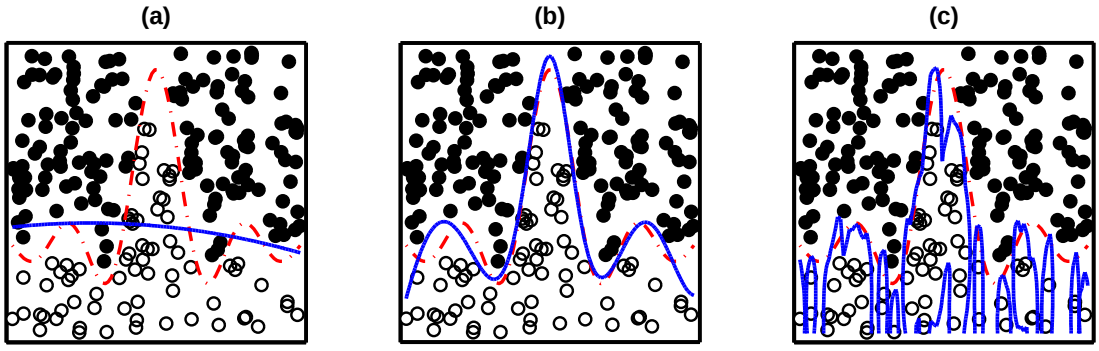

Figure 3.1: Empirical risk minimization and the over fitting phenomenon. Given the sample set consisting of two classes, a learning machine with limited capacity learns a simple function as shown by the solid decision boundary in (a), whereas a very flexible machine achieves zero training error (c) but fails to generalize to the true underlying function indicated by the dashed line. The optimal machine (b) trade-offs between capacity and minimizing training error.

key theorem of VC theory provides sufficient and necessary conditions for convergence of the ERM principle; any learning algorithm which is based on the ERM principle must satisfy it.

Theorem 3.1 (Asymptotic Consistency, (Vapnik and Chervonenkis, 1991)). *One-sided uniform convergence in probability,*

$$\lim_{N\to\infty} \mathcal{P}\left[\sup_{f\in\mathcal{H}} (\mathcal{R}(f) - \mathcal{R}_{emp}(f)) > \epsilon\right] = 0, \tag{3.4}$$

for all $\epsilon > 0$, is a necessary and sufficient condition for (nontrivial[3]) consistency of empirical risk minimization. The set of functions is assumed to have a bounded loss for some probability measure μ, that is

$$A \leq \int f \, d\mu \leq B, \text{ for all } f \in \mathcal{H}. \tag{3.5}$$

Theorem 3.1 asserts that the worst case over all functions that the learning machine can implement determines the consistency of ERM (Vapnik, 1999). Intuitively, the key learning theorem using the ERM principle requires one to choose f^* from the set of functions that satisfy the necessary and sufficient conditions. To this end, a notion of dimensionality or capacity of $\mathcal{H}$ which captures the complexity of functions in it is required. A simple measure of the complexity of a hypothesis space originally proposed in VC theory is the Vapnik-Chervonenkis dimension (h_d). The complexity metric h_d measures how many (training) points can be separated or *shattered* for all possible labellings using functions of the class. As an illustration of the concept, consider a binary classification problem in $\mathbb{R}^2$. Taking the set of linear separating hyperplanes as the hypothesis space, a maximum of three instances

[3] This requires removing atypical functions from the hypothesis space otherwise if there is an f^* which has the smallest error over all $f \in \mathcal{H}$ for all sample sizes m, the learning algorithm will always choose that function.

can be separated without error for all arbitrary labellings, Figure 3.2. However, a set of four points cannot be shattered by the same class. Hence, the shattering dimension h_d of $\mathbb{R}^2$ for the class of linear hyperplanes is three. In general, a maximum of $d+1$ points can be shattered by the class of hyperplanes for any $\mathbb{R}^d$.

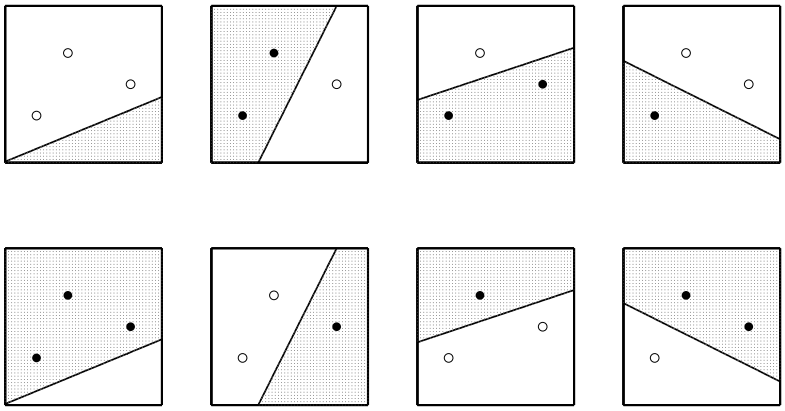

Figure 3.2: An illustration of the shattering dimension for $\mathbb{R}^2$ space for a class of linear separating hyperplanes. Here, filled circles indicate negative instances, and the open circles positive labels.

Generalization bounds using capacity metrics such as h_d can be derived to characterize the performance of a learning algorithm as

$$\mathcal{R}(f) \leq \mathcal{R}_{\text{emp}}(f, \mathcal{T})) + \mathcal{G}(\mathcal{H}, m, \eta), \quad \eta > 0 \tag{3.6}$$

where $\mathcal{G}$ is a confidence function, η a probability, $\mathcal{T}$ the training sample, and m the sample size. The generalization bound is a sum of the empirical error and a confidence term that depends on the hypothesis space from which f is chosen and the sample size of the training set. Ideally, to achieve some guarantee (up to some probability specified by η) that the actual risk or generalization error is small an induction principle is needed that minimizes both terms. In the case of a pattern recognition, an example of such a bound is defined as follows (Vapnik, 2000). Given some $0 \leq \eta \leq 1$, and for a 0/1 loss function the following bound on the functional risk

$$\mathcal{R}(f) \leq \mathcal{R}_{\text{emp}} + \sqrt{\frac{h_d(\log(2m/h_d)+1) - \log(\eta/4)}{m}} \tag{3.7}$$

holds with probability $1-\eta$ for $m > h_d$ over a random draw of the sample $\mathcal{T}$.

It must be noted that Equation (3.7) is independent of $\mathcal{P}(\mathcal{X}, \mathcal{Y})$ (all the information available concerning the generating distribution is the i.i.d. training data). Although the term $\mathcal{R}(f)$ may not be computable, the right hand side can be evaluated if h_d is specified. Therefore, given a hypothesis space $\mathcal{H}$, selecting an f that minimizes the right hand side gives an f with minimal upper bound on the expected loss up to some probability $1-\eta$. This motivates the induction principle of structural risk minimization (SRM) (Burges, 2004; Vapnik, 1979).

3.1.3 Structural Risk Minimization

The objective in SRM is to select a subset of the hypothesis space with the smallest possible confidence term, and a function f from that subset with minimal empirical error. Unfortunately, h_d is non-integral. To proceed, Vapnik suggested partitioning the class of functions into nested subsets $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \ldots \subset \mathcal{H}_k$ arranged such that $h_1 \leq h_2 \ldots \leq h_k$, with corresponding ERM solutions $f_1, f_2, \ldots, f_k$ in the function sub-classes $\mathcal{H}_i$ (Equation 3.3). The SRM principle then chooses the function class $\mathcal{H}_i$ with minimal upper bound on the generalization error (Equation 3.7). Figure 3.3 illustrates the idea.

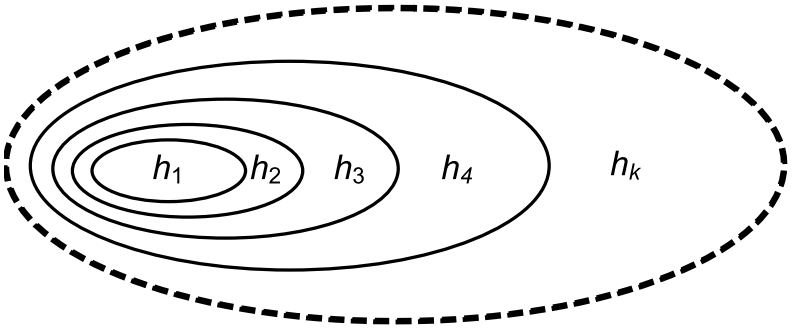

Figure 3.3: The structural risk minimization principle partitions the hypothesis space $\mathcal{H}$ such that $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \ldots \subset \mathcal{H}_k \subset \mathcal{H}$, and chooses an $f_i^* \in \mathcal{H}_i$ with minimal empirical risk, where $\mathcal{H}_i$ is the subclass with minimal generalization.

VC theory establishes generalization bounds using some metric (like h_d) on the hypothesis space from which the optimal function is chosen. It is possible to use other notions characterizing the complexity of a class of functions. For example, covering numbers which measure the number of balls of a given radius needed to cover the space of functions have been suggested (Williamson et al., 2001). Both VC-style dimensions and covering numbers define a metric on the function class. Rademacher averages have also been proposed that are directly related to the object of interest, that is, maximum of an empirical process (Boucheron et al., 2005; Bousquet, 2003). To extend the characterization result for any learning algorithm other than ERM, notions of uniform stability and cross validation leave-one-out stability have been suggested (Poggio et al., 2004). Instead of characterizing the hypothesis space, these are aimed at developing a general theory that characterizes the properties of a mapping that ensure good generalization error. These will not be discussed except to mention that the study of generalization bounds is intricately related to constructing practical learning algorithms with improved performance.

In the next sections practical contributions with respect to computational algorithms for learning based on the VC theory as outlined above are introduced. However, first some philosophical remarks underlying VC theory are in order.

3.1.4 Philosophical Remarks

The modern formulation of the induction problem, that is inference of a general theory on the basis of empirical facts, is attributed to David Hume, a Scottish philosopher. In his

"problem of induction", Hume (1777) conjectured that since all we know about nature is derived from our experience, to what extent was inductive inference justified? (A similar problem occurs in machine learning; given observed data, a learning machine processes it and outputs a prediction. What guarantees or confidence does one have that the prediction is correct?) In response, Karl Popper (1968), using a falsification framework[(4)], stated that while a scientific theory could not be justified, it made sense however to demarcate false and true theories. To this end, he proposed two methods for comparing theories:

1. The containment relation between classes of falsifiers.
2. The dimension of a theory characterizing how complex a theory is. Simpler theories are to be preferred over complicated theories.

Karl Popper's reasoning inspired modern statistical learning theory (Vapnik, 2000), particularly the insight that while it is impossible to build a general learning machine, the possibility existed of building a learning machine and theoretically analyze its performance *within a constrained framework*.

3.2 Supervised Learning

3.2.1 Large Margin Classification

The concept of large margin classification takes into account the necessary statistical learning bias in machine learning algorithms and is central in understanding the improved performance achieved by support vector machines and related algorithms. Assume a training sample $\mathcal{T}$ is given

$$\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}, \text{ with } (\mathbf{x}, y) \in \mathbb{R}^d \times \{-1, 1\} \tag{3.8}$$

and the objective is finding a linear decision function $f: \mathbb{R}^d \to \{-1, 1\}$

$$f(\mathbf{x}) = (\mathbf{x} \cdot \mathbf{w}) + b \tag{3.9}$$

where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are the weight and bias parameters respectively, such that the classification error $(\text{sgn}(f(\mathbf{x})) \neq y)$ is minimized. The decision function or hyperplane separates the input space into half-spaces where sample points on one side of the hyperplanes are assigned into one class (+1) while the others are assigned into the other class (-1). The margin corresponding to a specific f is defined by

$$\rho_f := yf(\mathbf{x}) \tag{3.10}$$

and quantifies the amount by which $\mathbf{x}$ is classified correctly. A negative margin value will therefore indicate an incorrect classification, as well as by how much f came close to choosing the alternative label. With a slight abuse of notation, denoting the minimum margin over $\mathcal{T}$ by

$$\rho_f := \min_{i \in \mathcal{T}} y_i f(\mathbf{x}_i), \tag{3.11}$$

(4) Falsification is the existence of a collection of particular assertions that cannot be explained by a given theory although they fall into its domain.

it can be expected that an f with large margin ρ_f on the training set will perform well on test examples. Therefore, the binary pattern recognition problem can be re-formulated as finding an f^* with maximum margin, that is[5]

$$\begin{aligned} f^* &:= \underset{f}{\operatorname{argmax}}\, \rho_f \\ &= \underset{f}{\operatorname{argmax}} \min_{i \in \mathcal{T}} y_i f(\mathbf{x}_i). \end{aligned} \tag{3.12}$$

Now, a unique solution to Equation (3.12) does not exist since for any parameterization of the hyperplane $(\mathbf{w}, b)$ in Equation (3.9), any $\alpha \neq 0$, the parameters $(\alpha\mathbf{w}, \alpha b)$ describe the same hyperplane, that is

$$\{\mathbf{x} \,|\, (\mathbf{w} \cdot \mathbf{x}) + b = 0\} \equiv \{\mathbf{x} \,|\, (\alpha\mathbf{w} \cdot \mathbf{x}) + \alpha b = 0\}. \tag{3.13}$$

Unique representation can be obtained by requiring correspondence between the geometrical hyperplane and the parameterization. This can be achieved by scaling $(\mathbf{w}, b)$ such that

$$f(\mathbf{x}) = ((\mathbf{w} \cdot \mathbf{x}) + b)/\|\mathbf{w}\|.$$

Such a normalized hyperplane is called a *canonical hyperplane*. The maximal margin hyperplane parameters are then defined by

$$(\mathbf{w}^*, b^*) = \underset{\mathbf{w}, b}{\operatorname{argmax}} \left\{ \min_{i \in \mathcal{T}} \frac{y_i((\mathbf{w} \cdot \mathbf{x}_i)) + b}{\|\mathbf{w}\|} \right\} \tag{3.14}$$

$$= \underset{\mathbf{w}, b}{\operatorname{argmax}} \left\{ \min_{i \in \mathcal{T}} \left(y_i \operatorname{sgn}(\mathbf{w} \cdot \mathbf{x}_i) + b \left\| \frac{(\mathbf{w} \cdot \mathbf{x}_i)}{\|\mathbf{w}\|^2} \mathbf{w} + \frac{b}{\|\mathbf{w}\|^2} \mathbf{w} \right\| \right) \right\}. \tag{3.15}$$

Geometrically, the term $-b\mathbf{w}/\|\mathbf{w}\|^2$ in Equation (3.15) is the vector in direction $\mathbf{w}$ that terminates on the decision hyperplane, and $(\mathbf{w} \cdot \mathbf{x}_i)\mathbf{w}/\|\mathbf{w}\|^2$ is the projection of $\mathbf{x}_i$ onto $\mathbf{w}$. Hence, an optimal margin hyperplane is the one that maximizes the vector differences $((\mathbf{w} \cdot \mathbf{x}_i)\mathbf{w}/\|\mathbf{w}\|^2 - (-b\mathbf{w}/\|\mathbf{w}\|^2)$. Introducing a lower bound on the margin ρ, Equation (3.14) can be transformed into the following optimization problem;

$$(\mathbf{w}^*, b^*, \rho^*) = \underset{\mathbf{w}, b, \rho}{\operatorname{argmax}} \quad \rho \tag{3.16}$$

$$\text{subject to} \quad \frac{y_i((\mathbf{w} \cdot \mathbf{x}_i) + b)}{\|\mathbf{w}\|} \geq \rho, i = 1, .., m \tag{3.17}$$

Noting that the weight vector is in canonical form, Equations (3.16)–(3.17) can be formulated as

$$(\mathbf{w}^*, b^*, \rho^*) = \underset{\mathbf{w}, b, \rho}{\operatorname{argmax}} \quad \rho \tag{3.18}$$

$$\text{subject to} \quad \begin{cases} \|\mathbf{w}\| \text{ and,} \\ y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq \rho, i = 1, .., m \end{cases} \tag{3.19}$$

[5] argmax is the maximum of the given argument for which the expression attains its maximum value, i.e.

$$\underset{x}{\operatorname{argmax}}\, g(x) \in \{x | z \neq x \implies g(z) < g(x), \text{ for all } z\}$$

which is equivalent to

$$(\mathbf{w}^*, b^*, \rho^*) = \underset{\mathrm{w}, b}{\operatorname{argmin}} \quad \|\mathbf{w}\|^2 \tag{3.20}$$

$$\text{subject to} \qquad y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq \rho, i = 1, .., m. \tag{3.21}$$

From Equations (3.18) and (3.19) it can be seen that a weight vector $\mathbf{w}$ is desired, with large dot products $y_i(\mathbf{w} \cdot \mathbf{x}_i)$ constrained to lie on the unit sphere. Computationally, the quadratic optimization form in Equations (3.20) and (3.21) can be efficiently solved using standard optimizers. Moreover, the quadratic form permits other ways of constraining the unit sphere besides ℓ_2-margin, for example ℓ_1-margin, ℓ_∞-margin, or more generally ℓ_p-margins (Bradley, 1998; Mangasarian, 1997; Smola et al., 2000). Finding the solution by optimizing over the margin ρ_f corresponds to using a convex surrogate loss for the 0/1 loss function (risk convexification). This avoids solving a potentially intractable problem for most nontrivial function classes. Besides the computational advantage, large margin classification is an implicit form of regularization (Bartlett, 1998; Schapire et al., 1998; Vert et al., 2005). Furthermore, different functional forms of the convex surrogate loss can be used, for example hinge loss (used in standard SVM algorithm), exponential loss (used in boosting algorithm), and the logit loss.

The following theorem due to Vapnik (1998) explains why large margin classifiers should be expected to perform well despite the high-dimensionality of the associated feature spaces.

Theorem 3.2. *Let $X = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\mathrm{m}$ be a set of vectors belong to the smallest sphere of radius R centered on the origin. The VC dimension of the set of (canonical) hyperplanes $\{f: X \to [-1, 1] | f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}_i)\}$, $\|\mathbf{w}\| \leq \Lambda\}$ is bounded from above by the inequality*

$$h_d \leq \min(R^2\Lambda^2, d) + 1 \tag{3.22}$$

Hence, although the VC dimension of hyperplanes can be $d + 1$, where d is the dimension of the space, the VC dimension of a subclass (margin hyperplanes) can be much smaller. Also, by bounding the margin of the hyper class not to be smaller than some quantity (e.g. $2/\Lambda$) one can control the class complexity and hence apply the SRM induction principle. Also, the dimensionality of the space does not influence the learning. As discussed below, these insights play an important role in the development of support vector machine (SVM) algorithms. However, in the case of SVMs it is not necessary to specify the structure of the class of functions *a priori* via $\|\mathbf{w}\| \leq \Lambda$ (Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002).

It must be noted that the SRM principle alone does not adequately account for the good generalization performance of large margin classifiers because the resulting bounds are rather too loose, and other factors are needed to provide a rigorous explanation (Burges, 2004).

3.2.2 Support Vector Machines

In the preceding section the large margin VC theory bias that is necessary for good generalization performance of learning machines was introduced. The support vector machine

(Boser et al., 1992; Burges, 2004) is a learning algorithm developed using the foregoing insights; that is, the training procedure finds an optimal $f \in \mathcal{H}$ which minimizes an upper bound on the sum of the empirical risk and a capacity term, Equation (3.6). Again, consider a binary classification problem for separable data

$$\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_1, y_1)\}, \text{ with } \mathbf{x} \in \mathbb{R}^d, y_i \in \{-1, +1\}, \tag{3.23}$$

and the goal is to find a separating canonical hyperplane $(\mathbf{w}, b)$ such that the following are satisfied

$$(\mathbf{w} \cdot \mathbf{x}_i) + b = \begin{cases} \geq +1, & \text{if } y_i = +1 \\ \leq -1, & \text{if } y_i = -1. \end{cases} \tag{3.24}$$

As discussed previously, the optimal solution $(\mathbf{w}^*, b^*)$ is obtained by solving the optimization problem

$$\min_{\mathrm{w}, b} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{3.25}$$

$$\text{subject to} \quad y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1, \quad \text{for all } i = 1, \ldots, m. \tag{3.26}$$

Noting that in the case of canonical hyperplanes the margin is given by $\gamma = 2/\|\mathbf{w}\|$ (Section 3.2.1), Equation (3.24) is equivalent to maximizing the margin and, hence controlling the VC dimension of the class of functions.

To solve Equation (3.26), the problem is transformed into a computationally easily manageable form by introducing Lagrange multipliers α_i, for $i = 1, \ldots, m$ for each inequality constraint (Fletcher, 1989). The subsequent optimization problem requires the minimization of the Lagrangian

$$\min_{\mathrm{w}, b, \boldsymbol{\alpha}} \quad \mathscr{L}_P(\mathbf{w}, b, \alpha_i) := \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i y_i (\mathbf{w} \cdot \mathbf{x} + b) + \sum_{i=1}^{m} \alpha_i \tag{3.27}$$

$$\text{subject to} \quad \alpha_i \geq 0, \text{ for } i = 1, \ldots, m. \tag{3.28}$$

Since Equation (3.27) is a convex optimization problem, one can instead consider the corresponding dual formulation

$$\max_{\boldsymbol{\alpha}} \quad [\min_{\mathrm{w}, b} \mathscr{L}_P(\mathbf{w}, b, \boldsymbol{\alpha})] \tag{3.29}$$

$$\text{subject to} \quad \begin{cases} \alpha_i \geq 0, & \text{for } i = 1, \ldots, m; \\ \dfrac{\partial \mathscr{L}_P}{\partial \mathbf{w}} = 0, \\ \dfrac{\partial \mathscr{L}_P}{\partial b} = 0. \end{cases} \tag{3.30}$$

The constraints on the gradients of the partial derivatives with respect to $(\mathbf{w}, b)$ yield

$$\mathbf{w} = \sum_{i}^{m} \alpha_i y_i \mathbf{x}_i,$$
$$0 = \sum_{i}^{m} \alpha_i y_i. \tag{3.31}$$

Substituting Equation (3.31) in the primal Lagrangian (Equation 3.27), Equation (3.29) can be expressed as

$$\max_{\boldsymbol{\alpha}} \quad \mathscr{L}_D := \sum_{i}^{m} \alpha_i - \frac{1}{2} \sum_{i}^{m} \sum_{j}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j), \tag{3.32}$$

$$\text{subject to} \quad \begin{cases} \sum_{i}^{m} \alpha_i y_i = 0, \\ \alpha_i \geq 0, i = 1, \ldots, m. \end{cases} \tag{3.33}$$

In the case of separable training data the solution to Equation (3.32) is sparse since $\alpha_i = 0$ for many of the samples. The samples for which $\alpha_i > 0$ are called support vectors and lie on the margin hyperplane as illustrated in Figure 3.4. The offset factor b is found by exploiting the Karush-Kuhn-Tucker (KKT) "complementarity" conditions; that is, the product between the dual variables and constraints (Equation 3.26) vanish at the solution point (Fletcher, 1989), i.e.

$$\alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i) + b) - 1)) = 0, \qquad \text{for all } i = 1, \ldots, m. \tag{3.34}$$

Hence, the bias parameter b can be found by substituting any sample with $\alpha_i \neq 0$ in Equation (3.34). Alternatively, when using interior point optimization algorithms b is implicitly determined during the training (Schölkopf and Smola, 2002; Smola, 1998).

To extend the linear SVM algorithm to handle non-separable data, Cortes and Vapnik (1995) proposed to relax the constraints of the optimization problem by defining non-negative slack variables $\xi_i, i = 1, \ldots, m$ such that

$$y_i ((\mathbf{x}_i \cdot \mathbf{w}) + b) \geq 1 - \xi_i, \text{ and} \tag{3.35}$$
$$\xi_i \geq 0, \text{ for } i = 1, \ldots, m. \tag{3.36}$$

The slack variables ξ_i allow for classification errors during training, and the number of training errors is upper bounded by $\sum_{i=1}^{m} \xi_i$. The optimization problem is then

$$\min_{\mathrm{w}, b} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \left(\sum_{i}^{m} \xi_i \right)^{\varrho}, \tag{3.37}$$

$$\text{subject to} \quad y_i (\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1, \qquad \text{for all } i = 1, \ldots, m \tag{3.38}$$

where a C is a user-defined parameter specifying the penalty of misclassification and ϱ is a positive integer, usually chosen to be $\varrho = \{1, 2\}$. The choice $\varrho = 1$ is particularly appealing

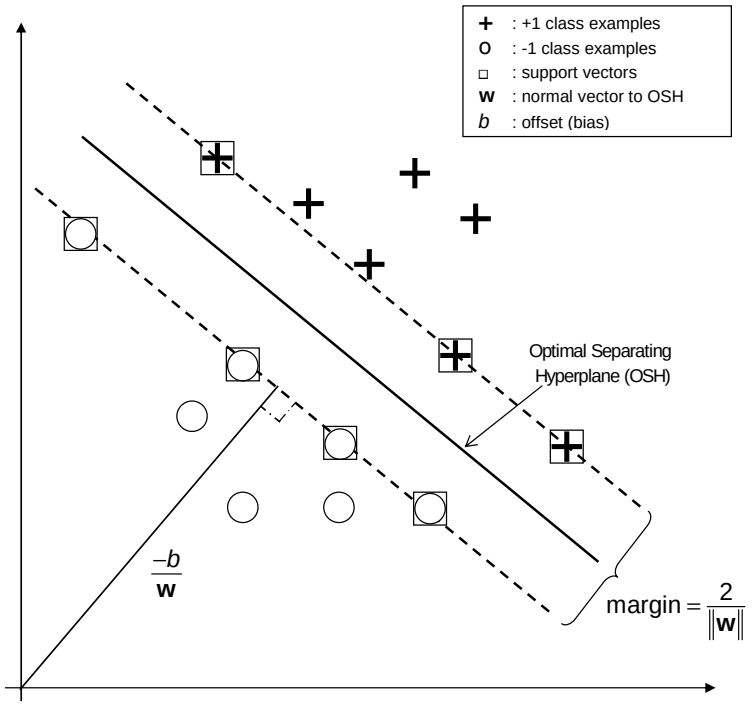

Figure 3.4: Geometric characterization of a linearly separable support vector classification problem.

from an optimization viewpoint because the slack variables ξ_i and the associated Lagrange multipliers disappear in the corresponding dual formulation,

$$\max_{\boldsymbol{\alpha}} \mathscr{L}_D := \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x} \cdot \mathbf{x}), \tag{3.39}$$

$$\text{subject to} \begin{cases} 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m; \\ \sum_{i=1}^{m} \alpha_i y_i = 0. \end{cases} \tag{3.40}$$

Because Equation (3.39) is similar to the separable dual formulation, it has the same solution except for an upper bound C on the penalization error term in the former.

The parameter C is rather unintuitive and cannot be specified automatically but determined via an exhaustive search. Schölkopf et al. (2000) suggested a different parameterization which replaces C by a parameter ν that controls the level of acceptable error in solving the classification problem. The resulting classification problem is termed ν-SVC. The ν-SVC primal optimization problem is expressed as

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 - m\nu\rho + \sum_{i=1}^{m} \xi_i, \tag{3.41}$$

$$\text{subject to} \begin{cases} y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq \rho - \xi_i, \\ \xi_i \geq 0, \text{ for } i = 1, \ldots, m, \\ \rho \geq 0, \end{cases} \tag{3.42}$$

where the fraction of training points with $\xi_i > 0$ (called margin errors) is $1/m \sum_i I(y_i \mathbf{x}_i < \rho)$. The corresponding dual formulation is given by

$$\max_{\boldsymbol{\alpha}} \qquad \mathscr{L}_D := -\frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{3.43}$$

$$\text{subject to} \quad \begin{cases} 0 \leq \alpha_i \leq \dfrac{1}{m}, \text{ for } i = 1, \ldots, m, \\ \displaystyle\sum_{i=1}^{m} \alpha_i y_i = 0, \\ \displaystyle\sum_{i=1}^{m} \alpha_i \geq \nu. \end{cases} \tag{3.44}$$

The parameter ν simultaneously upper bounds the fraction of margin errors and lower bounds the fraction of support vectors. Also, ν equals both the fraction of support vectors and the fraction of errors as $m \to \infty$ (Schölkopf et al., 2000). A connection between C-SVMs and ν-SVMs has been described by Schölkopf et al. (2000) who noted that the ν-SVC classifier with $\rho > 0$ and a C-SVM classifier with an a priori specified $C = 1/\rho$ share the same solution.

3.2.3 Kernel Functions

Although linear SVM algorithms provide a statistically principled approach to learning from data, they are limited in practical applications where flexible methods that can handle nonlinearity are required. As in classical linear modelling, one can introduce nonlinearity by pre-processing the data by some nonlinear function before learning a decision function, that is

$$\begin{aligned} \boldsymbol{\phi}: \quad \mathcal{X} &\to \mathcal{H} \\ \mathbf{x} &\mapsto \boldsymbol{\phi}(\mathbf{x}) \end{aligned} \tag{3.45}$$

where $\mathcal{H}$ is some *feature space*. For example, given data in $\mathbb{R}^2$ space and assuming most information is contained in 2^{nd} order products of vector entries $(x_i \cdot x_j)$ for $i, j = (1, 2)$, it maybe preferable to work in $\mathbb{R}^3$ feature space given by the mapping

$$\begin{aligned} \boldsymbol{\phi}: \quad \mathbb{R}^2 &\to \mathbb{R}^3 \\ (x_1, x_2) &\mapsto (x_1^2, \sqrt{2} x_1 x_2, x_2^2). \end{aligned} \tag{3.46}$$

Here, the term $\sqrt{2}$ compensates for number of occurrences of the term $x_1 x_2$ (Schölkopf, 1997). Generalizing to n-dimensional inputs, it can be shown that extracting p-th order monomials gives a feature space of dimensionality (Schölkopf, 1997; Schölkopf and Smola, 2002)

$$\dim(\mathcal{H}) = \binom{n+p-1}{p} \tag{3.47}$$

$$= \frac{(n+p-1)!}{p!(n-1)!}. \tag{3.48}$$

For large n the problem degenerates into a combinatorial large dimensional feature space

and, consequently, a computationally complex problem to solve. For example, for $2^4 \times 2^4$ dimensions and $p = 5$ the feature space is of dimension of the order 10^{10}!

Nonlinear SVMs and related kernel-based methods are based on the observation that the training data only appear as inner or dot products in the dual optimization algorithms, Equations (3.29), (3.39), and (3.43) (Boser et al., 1992). In other words, one never works directly with the feature space representation except via the dot products[6]. Thus, pre-processing the data using Equation (3.45) implies the transformed data appears in the learning algorithm as pairwise comparisons $\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle$ in feature space $\mathcal{H}$. The combinatorial problem associated with working in high-dimensional spaces can be circumvented if computation of the dot products can be done implicitly via a function $\mathbf{k}$ such that

$$\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle. \tag{3.49}$$

This raises the question of what functions $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$ exist that are equivalent to computing dot products in a high-dimensional feature space $\mathcal{H}$. Mercer's theorem guarantees that the so-called *Mercer kernels* evaluated on data in input space $\mathcal{X}$ correspond to computing dot products in $\mathcal{H}$ (Boser et al., 1992; Burges, 2004; Schölkopf and Smola, 2002; Vapnik, 2000). Mercer's condition is formally stated as follows.

Theorem 3.3 (Mercer's Theorem)**.** *Given a continuous symmetric kernel* $\mathbf{k}$ *of a positive integral operator* $\mathbf{K}$ *such that* $(\mathbf{K}f)(\mathbf{y}) = \int_{\mathcal{X}} \mathbf{k}(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) d\mathbf{x}$ *is positive definite,*

$$\int \mathbf{k}(\mathbf{x}, \mathbf{x}') f(\mathbf{x}) f(\mathbf{x}') d\mathbf{x}\, d\mathbf{x}' \geq 0 \tag{3.50}$$

for all $f \in L_2(\mathcal{X})$ *(where* $\mathcal{X}$ *compact), it can be expanded in a uniformly convergent series on* $\mathcal{X} \times \mathcal{X}$ *in terms of normalized eigenfunctions* ψ_i *with* λ_i *the corresponding eigenvalues*

$$\int \mathbf{k}(\mathbf{x}, \mathbf{y}') = \sum_{i=1}^{N_F \leq \infty} \lambda_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y}). \tag{3.51}$$

Thus for any $\mathbf{k}$ satisfying Theorem 3.3 the relationship in Equation (3.49) holds.

As an illustration of how to think of the effect of a mapping function on some input space, consider data defined on a square $[-1, 1] \times [-1, 1] \in \mathbb{R}^2$. Assuming most of the information is contained in 2^{nd} order monomials (independent of the ordering), the $\mathbb{R}^3$ *features* are extracted via the mapping [7]

$$\begin{aligned} \boldsymbol{\phi}: \quad \mathbb{R}^2 &\rightarrow \mathbb{R}^3 \\ (x_1, x_2) &\mapsto (x_1^2, \sqrt{2} x_1 x_2, x_2^2). \end{aligned} \tag{3.52}$$

[6] Recall the definition of a dot product:
Definition 1 (Dot Product)**.** *Given a vector space* $\mathcal{X}$*, a dot product is a mapping* $\langle \cdot, \cdot \rangle$ *with* $\mathcal{X} \otimes \mathcal{X} \rightarrow \mathbb{R}$ *which for all* $\alpha \in \mathbb{R}$ *and* $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}$ *satisfies*

1. $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$ *(symmetry)*
2. $\langle \mathbf{x}, \alpha \mathbf{y} \rangle = \alpha \langle \mathbf{x}, \mathbf{y} \rangle$ *(linearity)*
3. $\langle \mathbf{x}, \mathbf{y} + \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{z} \rangle$ *(additivity)*

[7] Note the mapping and the feature space are not uniquely defined (Burges, 2004)

As shown in Figure 3.5, the transformation has the effect of warping the data in a high dimensional $\mathbb{R}^3$ feature space, although the intrinsic dimension of the data is $\mathbb{R}^2$ (Burges, 2004).

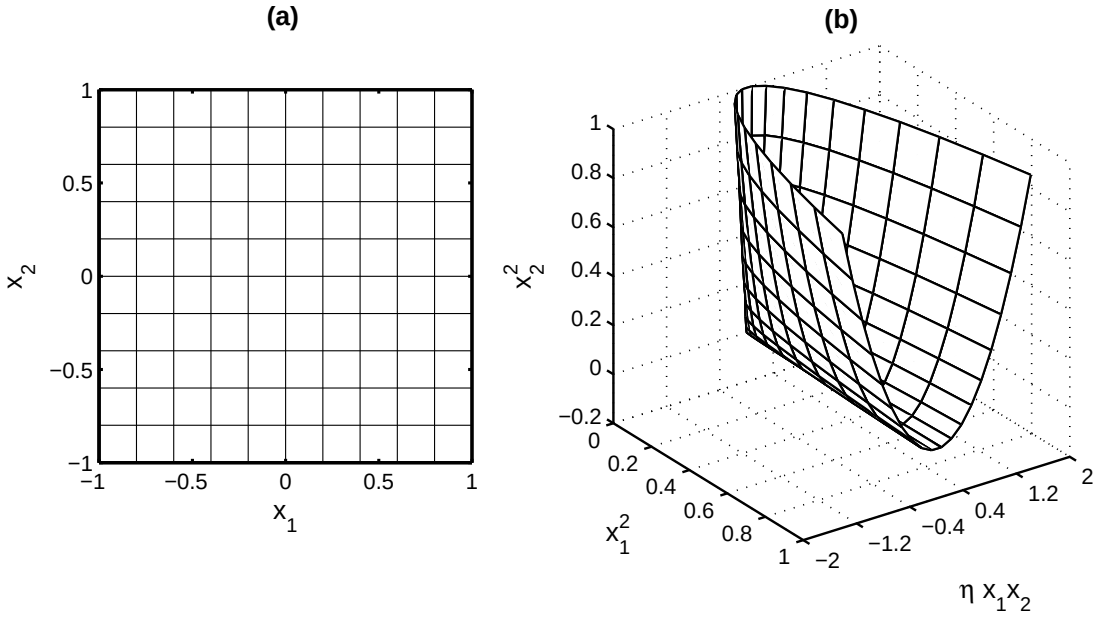

Figure 3.5: (a) Input space $\mathbb{R}^2$ and (b) resulting image when pre-processed by $\boldsymbol{\phi}$.

More generally, for any positive definite kernel $\mathbf{k}$ one can define a mapping $\boldsymbol{\phi}$ into a feature space $\mathcal{H}_k$ such that $\mathbf{k}$ computes the dot product under the image of $\boldsymbol{\phi}$. Positive definite kernels are kernels which are symmetric and satisfy

$$\sum_{i,j} a_i a_j \mathbf{K}_{ij} \geq 0 \tag{3.53}$$

for any $a_1, \ldots, a_m \in \mathbb{R}$, where $\mathbf{K}_{ij}$ is the kernel matrix entry corresponding to $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$. Moreover, positive kernels satisfy $\mathbf{K}_{ii} \geq 0$.

The feature space associated with positive definite kernels can be constructed as follows. Given a positive definite kernel $\mathbf{k}$, a data set $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subseteq \mathcal{X}$ define $\mathcal{H}_k$ as the set of functions $f : \mathcal{X} \to \mathbb{R}$. Let $\boldsymbol{\phi}$ be the mapping

$$\begin{aligned} \boldsymbol{\phi} : \quad \mathcal{X} &\to \mathbb{R}^{\mathcal{X}} \\ \mathbf{x} &\mapsto \mathbf{k}(\cdot, \mathbf{x}) \end{aligned} \tag{3.54}$$

where $\mathbb{R}^{\mathcal{X}}$ is the space of all functions mapping $\mathcal{X}$ into $\mathbb{R}$. The image of $\mathcal{X}$ under $\boldsymbol{\phi}$ can be turned into a linear space by taking linear combinations

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i \mathbf{k}(\cdot, \mathbf{x}_i). \tag{3.55}$$

Given two functions of $\mathcal{H}_k$:

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i \mathbf{k}(\cdot, \mathbf{x}_i), \text{ and}$$
$$g(\cdot) = \sum_{j=1}^{m'} \beta_j \mathbf{k}(\cdot, \mathbf{x}'_i),$$

the dot product can be found as

$$\langle f, g \rangle = \sum_{i,j=1}^{m,m'} \alpha_i \alpha_j \mathbf{k}(\mathbf{x}_i, \mathbf{x}'_j). \tag{3.56}$$

Hence, $\mathcal{H}_k$ is a dot product space. The Hilbert space is obtained by completion of the norm by adding all the limit points under the norm induced by the kernel, that is

$$\|f\|^2_{\mathcal{H}_k} = \sum_{i,j=1}^{m} \alpha_i \alpha_j \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j). \tag{3.57}$$

Interesting properties follow from the foregoing construction of $\mathcal{H}$. The value of a function $f \in \mathcal{H}_k$ at a point $\mathbf{x}$ can be expressed as a dot product in the feature space

$$f(\mathbf{x}) = \langle f, \mathbf{k}(\mathbf{x}, \cdot) \rangle_{\mathcal{H}_k}. \tag{3.58}$$

In particular, we have

$$\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle = \langle \mathbf{k}(\cdot, \mathbf{x}_j), \mathbf{k}(\cdot, \mathbf{x}_i) \rangle_{\mathcal{H}_k} \tag{3.59}$$
$$= \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j). \tag{3.60}$$

Because of the last property $\mathbf{k}$ is sometimes referred to as the *reproducing kernel* – taking the dot product of the kernel with itself recovers the kernel again. The corresponding space of functions $\mathcal{H}_k$ is called the reproducing kernel Hilbert space (RKHS) associated with $\mathbf{k}$ (Schölkopf and Smola, 2002) formally defined as follows.

Definition 2. *Let $\mathcal{X}$ be a nonempty set and $\mathcal{H}$ a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$. Then $\mathcal{H}$ is called a reproducing kernel Hilbert space endowed with the dot product $\langle \cdot, \cdot \rangle$ if there exists a function $\mathbf{k} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ with the properties that*

1. $\mathbf{k}$ *has the reproducing property $\langle f, \mathbf{k}(\cdot, \mathbf{x}) \rangle = f(\mathbf{x})$ for all $f \in \mathcal{H}$; in particular*

   $$\langle \mathbf{k}(\cdot, \mathbf{x}_j), \mathbf{k}(\cdot, \mathbf{x}_i) \rangle_{\mathcal{H}} = \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j);$$

2. $\mathbf{k}$ *spans $\mathcal{H}$, that is $\mathcal{H} = \overline{\text{span}\{\mathbf{k}(x, \cdot) | \mathbf{x} \in \mathcal{X}\}}$, where $\overline{\{\cdot\}}$ denotes completion of the argument.*

The RKHS as defined uniquely determines the kernel $\mathbf{k}$ (Schölkopf and Smola, 2002). Therefore, instead of explicitly specifying the mapping function $\boldsymbol{\phi}$ in Equation (3.45), use of a kernel function satisfying Mercer's condition is enough to guarantee the existence of an appropriate Hilbert space. Thus, in applying a learning algorithm in feature space, the exact formulation of the mapping function $\boldsymbol{\phi}$ is not necessary – choosing an appropriate $\mathbf{k}$ is sufficient. Table 3.1 lists some commonly used kernel functions.

Table 3.1: Examples of commonly used Mercer kernel functions

Name	Functional Form
Homogeneous polynomial	$(\mathbf{x} \cdot \mathbf{y})^d, \quad d \in \mathbb{N}$
Inhomogeneous polynomial	$(\mathbf{x} \cdot \mathbf{y} + \theta)^d, \quad d \in \mathbb{N}$ and $\theta \geq 0$
Gaussian radial basis function	$\exp\left(-\|\mathbf{x} - \mathbf{y}\|^2 / 2\sigma^2\right), \quad \sigma > 0$
Sigmoid*	$\tanh(\kappa \mathbf{x} \cdot \mathbf{y} + \vartheta), \quad \kappa > 0$ and $\vartheta < 0$

* This is equivalent to a specific two-layer multilayer perceptron and is positive definite only for certain values of the hyperparameters (Vapnik, 2000)

The idea of performing operations in Hilbert space is not a novel one; it has been known in the mathematical sciences for a long time (see, e.g. Aronszajn (1950)). Although the idea of using kernels in machine learning applications was first proposed in Aizerman et al. (1964) (who used it in a convergence proof for the linear perceptron), the insight that it can be used in a mathematical programming setup is central in the development of powerful algorithms such as support vector machines (Boser et al., 1992), kernel discriminant analysis (Baudat and Anouar, 2000; Mika et al., 1999), kernel principal component analysis (Schölkopf et al., 1998), and density estimation (Girolami, 2002; Mukherjee and Vapnik, 1999) among other. Kernel functions are also used in Gaussian processes, a family of algorithms in machine learning closely related to support vector algorithms, where they are known as covariance operators (Rasmussen and Williams, 2006). The use of kernels has been extended to handle non-vectorial data such as strings in natural language processing (Haussler, 1999; Joachims, 1998; Watkins, 2000), graph kernels (Gärtner, 2003; Kashima et al., 2004), and tree kernels (Collins and Duffy, 2002).

Kernels as regularization operators

The RKHS representation of the feature space permits a useful interpretation of SVMs and other kernel methods from a functional analysis viewpoint. In regularization theory, instead of minimizing the upper bound on the empirical risk and a capacity term (Equation 3.6), one minimizes a *regularized risk* (Poggio and Girosi, 1990; Tikhonov and Arsenin, 1977),

$$\mathcal{R}_{\text{reg}}(f) = \mathcal{R}_{\text{emp}}(f, Z) + \frac{\lambda}{2}\|f\|_{\mathcal{H}} \tag{3.61}$$

over the entire space $\mathcal{H}$, where $\mathcal{R}_{\text{emp}}(f, Z)$ is small when f fits the data well. The norm $\|f\|_{\mathcal{H}}$ ensures a "smooth" solution which prevents overfitting (Schölkopf and Smola, 2002; Vert et al., 2004).

The representer theorem, originally due to Kimeldorf and Wahba (1971) and generalized in Schölkopf et al. (2000a), states that the solutions of certain risk minimization problems involving regularized risks of the form in Equation (3.61) can be expanded in terms of the training sample mapped into feature space even though the optimization is carried over a potentially infinite dimensional space:

Theorem 3.4 (Representer Theorem (Kimeldorf and Wahba, 1971))**.** *Let* $\Omega : [0, \infty] \to \mathbb{R}$ *be a strictly monotonic increasing function,* $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$ *and* L*:*$(\mathcal{X} \times \mathbb{R}^2) \to \mathbb{R} \bigcup \infty$ *an*

arbitrary loss function. Then each minimizing function $f \in \mathcal{H}$ of the regularized risk

$$L((\mathbf{x}_1, y_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_m, y_m, f(\mathbf{x}_m))) + \Omega(\|f\|_{\mathcal{H}}) \tag{3.62}$$

admits a representation of the form

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \mathbf{k}(\mathbf{x}_i, \mathbf{x}), \quad \textit{for all } \mathbf{x} \in \mathcal{X}. \tag{3.63}$$

The computational advantage arising from the representer theorem is that the optimal solution to Equation (3.62) can be obtained by re-formulating the problem as an m-dimensional optimization problem by substituting Equation (3.63) and expressing the solution in terms of $\alpha_i, \ldots, \alpha_m$ (Vert et al., 2004).

Nonlinear Support Vector Machines

In the preceding sections, the key ideas underlying support vector machines have been outlined, namely (i) a learning bias (large margin) from statistical learning theory and (ii) a method for implicitly evaluating dot products in feature spaces. In this section it is shown how to extend the flexibility of the linear SVMs using the kernel trick for both separable and non-separable pattern recognition problems, that is Equations (3.32) and (3.39). This is simple to achieve since the data appear only as dot products in both algorithms. Hence, the occurrence of each inner product is substituted with a kernel function, for example the Gaussian kernel (Table 3.1). This corresponds to pre-processing the data using a nonlinear mapping $\boldsymbol{\phi} : \mathbf{x} \to \boldsymbol{\phi}(\mathbf{x})$ and subsequently learning a linear function in the feature space $\mathcal{H}$. The choice of the kernel induces nonlinearity in input space.

Replacing all occurrences of dot product with a kernel function between two data points, the equivalent nonlinear formulation of the linearly separable (hard margin) dual optimization problem in Equation (3.32) is

$$\max_{\boldsymbol{\alpha}} \quad \mathscr{L}_D := -\frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) \tag{3.64}$$

$$\text{subject to} \quad \begin{cases} \sum_{i}^{m} \alpha_i y_i = 0, \\ \alpha_i \geq 0, i = 1, \ldots, m. \end{cases} \tag{3.65}$$

Similarly, the nonlinear soft margin equivalent of Equation (3.39) is

$$\max_{\boldsymbol{\alpha}} \quad \mathscr{L}_D := \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) \tag{3.66}$$

$$\text{subject to} \quad \begin{cases} 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m, \\ \sum_{i=1}^{m} \alpha_i y_i = 0. \end{cases} \tag{3.67}$$

In both cases the decision function for a test point $\mathbf{x}$ is obtained as

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i \in SVs} \alpha_i \mathbf{k}(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{3.68}$$

where SVs is the set of support vectors, that is training points with $\alpha_i > 0$. The bias parameter is obtained by exploiting the equivalent KKT conditions (Equation 3.34) or as solution to interior point optimization algorithm. In Appendix B.1 is sample MATLAB® code that implements a basic support vector algorithm for solving a binary pattern recognition problem.

Use of kernels circumvents the need to explicitly know the mapping $\boldsymbol{\phi}$ or feature space $\mathcal{H}$ except that it is vector space. By using geometrical concepts of angles and distances, kernel representation reduces otherwise complex nonlinear algorithms in $\mathcal{X}$ to simple linear formulations in $\mathcal{H}$. This insight is summed up as the "kernel trick": *any algorithm in which data appears as dot products can be implicitly performed in $\mathcal{H}$ by using kernel functions which permits the design of nonlinear versions of linear algorithms using rich function classes in input spaces* (Müller et al., 2001; Schölkopf and Smola, 2002). Figure 3.6 illustrates some of the learning properties of SVMs using different models and hyperparameter specifications in the case of a Gaussian kernel.

3.2.4 Discriminant analysis

Kernel-based nonlinear discriminant analysis

To extend the linear model in Equation (2.13) to the nonlinear case using kernel functions, a dot product formulation is required. In similar spirit to the SVM algorithm, the weight vector $\mathbf{w}$ can be expressed as a linear combination of the images of the training patterns under a mapping $\boldsymbol{\phi}$, that is

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \boldsymbol{\phi}(\mathbf{x}) \tag{3.69}$$

where α_i's are the expansion coefficients. Substituting Equation (3.69) for the weight vector in Equation (2.14) and after simplifying one obtains the feature space equivalent of the discriminant analysis optimization problem (assuming a binary system)

$$\max_{\boldsymbol{\alpha}} \vartheta(\boldsymbol{\alpha}) := \frac{(\boldsymbol{\alpha}'\boldsymbol{\mu})^2}{\boldsymbol{\alpha}' N \boldsymbol{\alpha}} = \frac{\boldsymbol{\alpha}' M \boldsymbol{\alpha}}{\boldsymbol{\alpha}' N \boldsymbol{\alpha}} \tag{3.70}$$

where $\boldsymbol{\alpha}_k = \mathbf{K}\mathbf{1}_k$, $N = \mathbf{K}(\mathbf{I} - \mathbf{1}_{N_k})\mathbf{K}'$, $\boldsymbol{\mu} = \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$, $M = \boldsymbol{\alpha}\boldsymbol{\alpha}'$, $(\mathbf{K}_i)_{pq} = \mathbf{k}(\mathbf{x}_p^i, \mathbf{x}_q^i)$ is the kernel matrix for class i, $\mathbf{1}_{N_k}$ an $N_k \times N_k$ matrix with all entries equal to $1/N_k$, and $\mathbf{I}$ the identity matrix. The solution to Equation (3.70) is obtained by solving a generalized eigenvector problem

$$M\boldsymbol{\alpha}_i = \lambda_i N \boldsymbol{\alpha}_i \tag{3.71}$$

where λ_i are the eigenvalues (Baudat and Anouar, 2000).

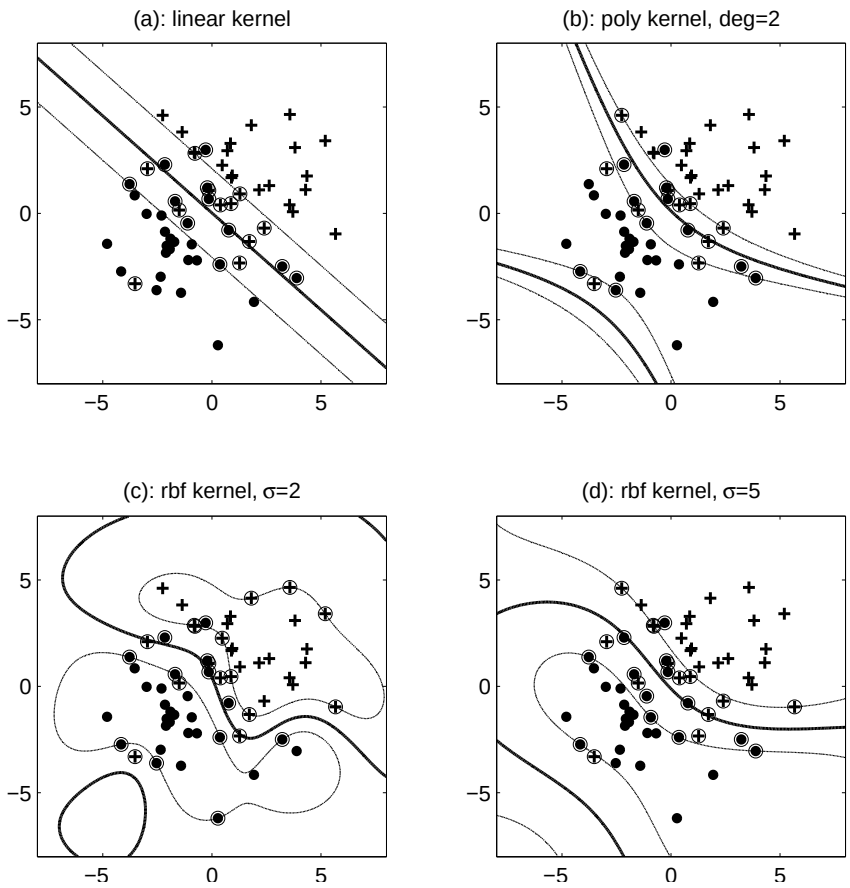

Figure 3.6: Soft margin SVM binary classification problem using (a) linear kernel (b) 2nd polynomial kernel, (c) an RBF kernel, width=2 and, (d) an RBF kernel, width=5. The regularization constant C was fixed at a value of 10 in all cases. The thick solid line is the decision boundary and the outer thin lines are the margins. The examples with non-zero α are indicated by a circle superimposed on the pattern. The support vectors shown by a circle are patterns falling within a margin, on margin boundaries, or incorrectly classified. See Appendix B.1 for self-contained MATLAB® code that implements this example.

Use of Equation (3.71) is limited to moderate m as M and N scale with the number of training points (Müller et al., 2001). Mika et al. (1999) showed that the nonlinear discriminant analysis problem in Equation (3.70) can be transformed into a convex optimization problem that can be simplified to a sparse-greedy approximation algorithm;

$$\min_{\boldsymbol{\alpha},b,\boldsymbol{\xi}} \quad \|\boldsymbol{\xi}\|^2 + C\Gamma(\boldsymbol{\alpha}) \tag{3.72}$$

$$\text{subject to} \quad \begin{cases} \mathbf{K}\boldsymbol{\alpha} + \mathbf{1}b = \mathbf{y} + \boldsymbol{\xi} \\ \mathbf{1}_i'\boldsymbol{\xi}, i = 1, 2 \end{cases} \tag{3.73}$$

where $\boldsymbol{\alpha}, \boldsymbol{\xi} \in \mathbb{R}^m$, $b, C \in \mathbb{R}$, Γ is a regularizer, and $(\mathbf{1}_i)_k$ is 1 if y_k belongs to class i and zero otherwise. The term $\boldsymbol{\xi}$ is an error term which takes into account the discrepancy induced by representing the feature space data in a reduced subset. Sample code imple-

menting the linear algebraic kernel-based FDA algorithm within the Spider machine learning environment[8] for MATLAB® is included in Appendix B.2.

3.3 Unsupervised Learning

The preceding section explored algorithms which take a sample dataset $\mathcal{T} = \{\mathbf{x}_i, y_i\}$, for $i = 1, \ldots, m$ where each input has an associated output or target value. Unsupervised learning is concerned with the case where the output values are not available. Although the unsupervised learning problem is less specified than the supervised version, it can generally be understood as aimed at understanding the process that generated the data. This is useful for many purposes such as extracting "interesting" features, data description, clustering, and density estimation.

3.3.1 Nonlinear Principal Component Analysis

In certain cases, principal components that are nonlinearly related to the original variables are of interest. However, PCA as described in Chapter 2 only considers first and second order moments. In the case where higher order moments are significant, alternative approaches are desirable for nonlinear feature extraction. A number of approaches have been proposed for nonlinear principal component analysis and these include principal curves (Hastie and Stuetzle, 1989) and artificial neural networks (Diamantaras and Kung, 1996; Kramer, 1992).

Recently, a generalization of linear PCA using kernel methods was proposed in Schölkopf et al. (1998). The general idea is to first map the data into some high-dimensional feature space $\boldsymbol{\phi} : \mathcal{X} \in \mathbb{R}^d \rightarrow \mathcal{H} \in \mathbb{R}^{n_h}$ and perform linear PCA (Equation 2.1) in that space. In some sense, the mapping seeks a suitable representation of the data in a high-dimensional (possibly infinite) $\mathbb{R}^{n_h}$ such that nonlinear features are unfolded. Let $\boldsymbol{\mu}_{\mathcal{H}}$ and $\mathbf{C}_{\mathcal{H}}$ be the respective empirical mean and covariance matrix of the image of the training set $\mathbf{X}$ under $\boldsymbol{\phi}$, that is

$$\boldsymbol{\mu}_{\mathcal{H}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{\phi}(\mathbf{x}_i), \tag{3.74}$$

$$\mathbf{C}_{\mathcal{H}} = \frac{1}{m}(\boldsymbol{\phi}(\mathbf{x}_i - \boldsymbol{\mu}_{\mathcal{H}})\boldsymbol{\phi}(\mathbf{x}_i - \boldsymbol{\mu}_{\mathcal{H}})'. \tag{3.75}$$

The eigen-decomposition problem in feature space is then

$$\mathbf{C}_{\mathcal{H}}\mathbf{w}_i = \lambda_i\mathbf{w}_i \text{ for all } i = 1, \ldots, m. \tag{3.76}$$

Before exploiting the kernel trick, Equation (3.76) needs to be expressed in terms of dot products between data points. In similar fashion to the formulation of the nonlinear discriminant analysis problem (Section 3.2.4) the principal loadings or eigenvectors can be expanded in terms of the mapped patterns: $\mathbf{w} = \sum_i \alpha_i\boldsymbol{\phi}(\mathbf{x}_i)$. Observing that all solutions $\mathbf{w}_i$ with non-zero eigenvalue λ_i lie in the span of the mapped training data, instead of

[8] Available at: http://www.kyb.tuebingen.mpg.de/bs/people/spider/

Equation (3.76) the following set of equations may be considered:

$$\lambda_i \langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}_i \rangle = \langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{C}\mathbf{w}_i \rangle, \text{ for all } i = 1, \ldots, m. \tag{3.77}$$

Substituting for $\mathbf{w}_i$ yields

$$\frac{1}{m} \sum_{j,j'=1}^{m} \langle \tilde{\boldsymbol{\phi}}(\mathbf{x}_i), \tilde{\boldsymbol{\phi}}(\mathbf{x}_j) \rangle \langle \tilde{\boldsymbol{\phi}}(\mathbf{x}_j), \tilde{\boldsymbol{\phi}}(\mathbf{x}_{j'}) \rangle \alpha_{j'} = \lambda_i \frac{1}{m} \sum_{j=1}^{m} \langle \tilde{\boldsymbol{\phi}}(\mathbf{x}_i), \tilde{\boldsymbol{\phi}}(\mathbf{x}_j) \rangle \tag{3.78}$$

where $\tilde{\boldsymbol{\phi}}(\mathbf{x}) := \boldsymbol{\phi}(\mathbf{x}) - \boldsymbol{\mu}_{\mathcal{H}}$. Since the data appear only in dot product terms, the kernel formulation of Equation (3.78) is therefore

$$\tilde{\mathbf{K}}^2 \boldsymbol{\alpha}_i = m\lambda_i \tilde{\mathbf{K}} \boldsymbol{\alpha}_i \tag{3.79}$$

where $\tilde{\mathbf{K}} = (\mathbf{I} - \mathbf{1}_m)\mathbf{K}(\mathbf{I} - \mathbf{1}_m)$, $\mathbf{K}_{ij} = \mathbf{k}((,)x_i, \mathbf{x}_j)$, $\mathbf{I}$ is the identity matrix, and $\mathbf{1}_m$ the matrix with all entries set to $1/m$. Absorbing m into λ, the eigen-system $(\lambda_i, \boldsymbol{\alpha}^i)$ is obtained as a solution to:

$$\lambda_i \boldsymbol{\alpha}_i = \mathbf{K}\boldsymbol{\alpha}_i. \tag{3.80}$$

The eigenvectors of $\mathbf{C}$ are given by

$$\mathbf{w}_i = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{m} \alpha_j^i \boldsymbol{\phi}(\mathbf{x}_i). \tag{3.81}$$

The projections of a test point $\mathbf{x}$ with image $\boldsymbol{\phi}(\mathbf{x})$ in $\mathcal{H}$ are evaluated according to

$$\begin{aligned} \langle \mathbf{w}_i, \boldsymbol{\phi}(\mathbf{x}) \rangle &= \frac{1}{\sqrt{\lambda_i}} \sum_{n=1}^{m} \alpha_n^i \langle \boldsymbol{\phi}(\mathbf{x}_n), \boldsymbol{\phi}(\mathbf{x}) \rangle \\ &= \frac{1}{\sqrt{\lambda_i}} \sum_{n=1}^{m} \alpha_n^i k(\mathbf{x}_n, \mathbf{x}). \end{aligned} \tag{3.82}$$

The scaling factor $1/\sqrt{\lambda_i}$ ensures orthonormality, that is $\langle \mathbf{w}_i, \mathbf{w}_i \rangle = 1$. As in the linear PCA, similar characterizations of kernel PCA apply (Section 2.3.2) with the exception that they become statements concerning feature space patterns $\boldsymbol{\phi}(\mathbf{x}_i)$, $i = 1, \ldots, m$ in $\mathcal{H}$ instead of $\mathbb{R}^d$ (Schölkopf and Smola, 2002; Schölkopf et al., 1998).

3.3.2 One-class Classification

Background and theoretical results

Conceptually, learning from unlabeled data is about estimating the density of an underlying probability distribution $\mathcal{P}$ generating the data. In theory, derivation of the density of $\mathcal{P}$ ($\mathrm{d}\mathcal{P}$) requires knowledge of $\mathcal{P}$, which must be continuous for $d\mathcal{P}$ to be well-defined (Vapnik, 2000). Unfortunately, in most cases the distribution is unknown and must be estimated from data. In many practical applications, it is not necessary to know the density of a distribution but the regions of space in which the mass of the data is concentrated, or support of the distribution. Estimation of the support of a distribution is considerably a

more tractable task than density estimation, particularly for finite-sized samples [9]. The goal is to find a function f that is positive in a "small" region capturing most of the data points, and negative elsewhere. In this sense, the learning task can be seen as a quantile (or minimum volume) estimation problem.

More formally, suppose a training set $\mathcal{T} = (\mathbf{x}_i, \ldots, \mathbf{x}_m)$ of i.i.d. random samples in input space $\mathcal{X} \in \mathbb{R}^d$ with distribution $\mathcal{P}$ is provided. Further, let $\mathcal{C}$ be a class of measurable subsets of $\mathcal{X}$, and let μ denote a real-valued function defined on $\mathcal{C}$. The α-quantile or simply quantile function with respect to $(\mathcal{P}, \mu, \mathcal{C})$ is (Polonik, 1997)

$$V(\alpha) \quad = \quad \inf\{\mu(C) : \mathcal{P}(C) \geq \alpha, C \in \mathcal{C}\}, \qquad 0 < \alpha \leq 1. \tag{3.83}$$

The empirical quantile function is defined as

$$V_m(\alpha) \quad = \quad \inf\{\mu(C) : \mathcal{P}_m(C) \geq \alpha, C \in \mathcal{C}\}, \qquad 0 < \alpha \leq 1 \tag{3.84}$$

where $\mathcal{P}_m := (1/m)\sum_i I_C(\mathbf{x}_i)$ is the empirical distribution of the training set samples and I_C is the indicator function on set C.

When μ is the Lebesgue measure, the solution set $C(\alpha)$ to Equation (3.83) is called the *minimum volume set* $C \in \mathcal{C}$ containing at least a fraction α of the probability mass. Similarly, the finite-sample solutions $C_m(\alpha)$ to Equation (3.84) are called minimum volume estimators.

A closely related task to minimum volume estimation is density level set estimation (DLSE). Instead of specifying the mass fraction α, DLSE methods enclose a region greater than a specified density level. As before, it is assumed a training set $\mathcal{T} = (\mathbf{x}_i, \ldots, \mathbf{x}_m)$ is available. Letting $\mathcal{P}$ be an unknown distribution on the input space $\mathcal{X} \in \mathbb{R}^d$, with density h with respect to a known reference measure μ. Given a desired density level $\rho > 0$, the goal of DLSE is to find the density level set $\Gamma(\rho) := \{h > \rho\}$ describing the concentration of $\mathcal{P}$ (Steinwart et al., 2005; Vert and Vert, 2006). In the general case, ρ-density level sets are similar to minimum volume sets for non-flat density function h:

$$\alpha \longleftrightarrow \rho, \qquad Q(\alpha) = \Gamma(\rho). \tag{3.85}$$

Algorithms for quantile estimation find a real-valued function $f{:}\mathcal{X} \to \mathbb{R}$ such that the set $\{f > 0\}$ is an estimate of the α-quantile $\{\mathcal{P}(C) > \alpha\}$. The one-class support vector machine algorithm for quantile estimation inspired by kernel-based methods is described next.

One-class support vector machine

The single or one-class support vector machine estimates the support of a distribution or region of space in which the data is concentrated by finding a hyperplane $\mathbf{w} \in \mathcal{H}$ that separates the unlabeled data $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ from the origin with maximum margin in feature space (Schölkopf et al., 2001). It has been shown that atypical objects in a set are concentrated around the origin in feature space (Twining and Taylor, 2003), thus the origin acts as a proxy for the unknown outlier class. The solution is found by solving the

[9] This is in spirit with Vapnik's principle – "Don't try to solve a problem by solving a harder one" (Vapnik, 1998).

quadratic programming problem,

$$\begin{aligned} \min_{\mathrm{x}\in\mathcal{H},\boldsymbol{\xi}\in\mathbb{R}^m,\rho\in\mathbb{R}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i - \rho \\ \text{subject to} \quad & \begin{cases} \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x})\rangle \geq \rho - \xi_i, \\ \xi_i \geq 0 \text{ for } i = 1, \ldots, m. \end{cases} \end{aligned} \tag{3.86}$$

The above optimization problem suggests retaining a large fraction of training patterns satisfying $f(\mathbf{x}) \geq \rho$, and simultaneously a small regularizer $\|\mathbf{w}\|^2$, with ν a trade-off parameter. Therefore, the decision function

$$f(\mathbf{x}) = \operatorname{sgn}\left((\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x})) - \rho\right) \tag{3.87}$$

will be positive in regions of high mass concentration, while the the regularizer $\|\mathbf{w}\|$ will still be smaller. Using tools from optimization theory, the Lagrangian of Equation (3.86) is

$$\begin{aligned} \mathscr{L}(\mathbf{w}, \boldsymbol{\xi}, \rho, \boldsymbol{\alpha}, \beta) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu m}\sum_{n}^{m}\xi_n - \rho \\ - \sum_{n}^{m}\alpha_n(\langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_n)\rangle - \rho + \xi_n) - \sum_{n}^{m}\beta_n\xi_n. \end{aligned} \tag{3.88}$$

At the saddle point of Equation (3.88), the primal variables $\mathbf{w}, \rho, \boldsymbol{\xi}$ are eliminated to obtain the dual optimization problem expressed only in terms of the Lagrangian multipliers,

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) \tag{3.89}$$

$$\text{subject to} \quad \begin{cases} 0 \leq \alpha_n \leq \dfrac{1}{\nu m} \\ \displaystyle\sum_{i=1}^{m}\alpha_i = 1. \end{cases} \tag{3.90}$$

The threshold term ρ is obtained using the Karush-Kuhn-Tucker conditions, that is, for any $0 \leq \alpha \leq 1/(\nu m)$, the corresponding pattern $\mathbf{x}$ satisfies

$$\rho = \langle \mathbf{x}, \boldsymbol{\phi}(\mathbf{x})\rangle = \sum_{i=1}^{m}\alpha_i\mathbf{k}(\mathbf{x}, \mathbf{x}_i). \tag{3.91}$$

Simple MATLAB® code implementing the one-class SVM of Schölkopf et al. (2001) is given in Appendix B.3.

It turns out that if the optimal solution of the one-class SVM model is $(\mathbf{w}, \rho)$, then $(\mathbf{w}, 0)$ is the corresponding optimal separating hyperplane for the binary classification problem using augmented data defined such that the original data belongs to, say, the positive class, and replicated and reflected images of the original data are assigned to the negative class,

$$\mathcal{T}_a := \{(\mathbf{x}_1, 1), \ldots, (\mathbf{x}_m, 1), (-\mathbf{x}_1, -1), \ldots, -(\mathbf{x}_m, -1)\}. \tag{3.92}$$

Hence, results derived within a binary classification context can be extended to single-class classification. For example, Schölkopf et al. (2001) used results derived for a ν-SVM classifier to characterize the trade-off parameter ν, namely that (i) ν upper bounds the number of outliers, (ii) ν is a lower bounds the number of patterns with $\alpha > 0$, and (iii) asymptotically ν equals the fraction of support vectors and fraction of outliers. Further theoretical statistical analysis including generalization bounds, consistency, and convergence issues of the one-class SVM are discussed in, for example Schölkopf et al. (2001); Vert (2006); Vert and Vert (2006).

Separating data from the origin with maximum margin in feature space is rather restrictive on the kind of outliers that can be detected by the algorithm presented above. To address this, several modifications to the one-class SVM have been proposed. For example, (Campbell and Bennett, 2001) proposed a linear programming (LP) algorithm that attracts the data toward the center of the data in feature space instead of maximizing the separation of data from an arbitrary point (the origin). The LP-based one-class SVMs also inspired development of one-class boosting algorithms that simplify the incorporation of prior knowledge into the problem setup (Rätsch et al., 2002).

Alternatively, prior information of what the "abnormal" class looks like can be encoded in the one-class SVM algorithm (Schölkopf et al., 2000b). In this case, the problem generalizes to finding a large margin hyperplane $\mathbf{w}$ that maximizes the separation of the set X from the centroid of another dataset $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$ with minimum training error. The set Z can be considered as modelling the distribution of the other "unknown" examples. More formally, the decision function is obtained by minimizing a weighted sum of a regularizer and a training error term that depends on an overall margin ρ and training errors ξ_i,

$$\min_{\mathrm{w}\in\mathcal{F}_{\mathcal{H}},\xi\in\mathbb{R}^{\ell},\rho\in\mathbb{R}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu\ell}\sum_{i=1}^{\ell}\xi_i - \rho \tag{3.93}$$

$$\text{subject to} \quad \langle \mathbf{w}, (\Phi(\mathbf{x}) - \frac{1}{t}\sum_{j=1}^{t}\mathbf{z}_j)\rangle \geq \rho - \xi_i, \quad \xi_i \geq 0. \tag{3.94}$$

The corresponding decision function is

$$f(\mathbf{x}) = \operatorname{sgn}\left(\left\langle \mathbf{w}, (\Phi(\mathbf{x}) - \frac{1}{t}\sum_{j=1}^{t}\mathbf{z}_j)\right\rangle - \rho\right) \tag{3.95}$$

which takes positive values for most patterns in X, while the regularizer $\|\mathbf{w}\|$ is still smaller. This is equivalent to a large margin of separation from the centroid of set Z.

As before, the dual form of Equation (3.93) is obtained by introducing the Lagrangian and expressing the the primal variables in terms of the Lagrange multipliers and the data. The primal variables are subsequently eliminated from the dual objective function which, in this particular case involves minimizing

$$\min_{\alpha\in\mathbb{R}^{l}} \quad \frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j\left(\mathbf{k}(\mathbf{x}_i,\mathbf{x}_j)\right) + Q - Q_j - Q_i \tag{3.96}$$

$$\text{subject to} \qquad 0 \leq \alpha_i \leq \tfrac{1}{\nu\ell}, \quad \sum_i \alpha_i = 1 \tag{3.97}$$

where $Q := \frac{1}{t^2}\sum_{np}\mathbf{k}(\mathbf{z}_n, \mathbf{z}_p)$ and $Q_i := \frac{1}{t}\sum_n \mathbf{k}(\mathbf{x}_i, \mathbf{z}_n)$. Finally, the function value of a test point $\mathbf{x}$ is evaluated according to

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_i \alpha_i \mathbf{k}(\mathbf{x}_i, \mathbf{x}) - \frac{1}{t}\sum_n \mathbf{k}(\mathbf{z}_n, \mathbf{x}) - \rho\right). \tag{3.98}$$

The threshold parameter ρ is computed by use of the Karush-Kuhn-Tucker (KKT) conditions. Thus, the feature extraction outputs large values for points similar to the image of $\mathbf{x}$ and small values for generic points from $\mathcal{Z}$. Specifying a *threshold* value such that if a novel point is drawn from the same distribution underlying $\mathcal{P}(\mathcal{X})$, it is possible to decide on whether or not it could have been generated from $\mathcal{P}(\mathcal{X})$ (Hayton et al., 2000; Schölkopf et al., 2000b).

A closely related algorithm to the one-class SVM algorithm is support vector data description (SVDD) that finds an optimal enclosing hypersphere with minimal volume (Tax, 2001; Tax and Duin, 1999). Formally, given the training set data $\mathcal{T}$ as before, the objective is to minimize the radius R of a ball centered on $\mathbf{c}$

$$\begin{aligned} \min_{R,\mathbf{c},\boldsymbol{\xi}} \quad & R^2 + C\sum_{i=1}^{m}\xi_i \\ \text{subject to} \quad & \begin{cases} \|(\mathbf{x}_i - \mathbf{c})\|^2 \le R^2 + \xi_i & \text{for } i = 1, \ldots, m \\ \xi_i \ge 0 \end{cases} \end{aligned} \tag{3.99}$$

where C is a trade-off parameter between minimization of the sphere volume and the number of outliers. The corresponding dual can be shown to be

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \sum_{i,j=1}^{m}\alpha_i\alpha_j(\mathbf{x}_i \cdot \mathbf{x}_j) - \sum_{i=1}^{m}\alpha_i(\mathbf{x}_i \cdot \mathbf{x}_i) \\ \text{subject to} \quad & 0 \le \alpha_i \le C, \ \sum_{i}^{m}\alpha_i = 1. \end{aligned} \tag{3.100}$$

The decision function of Equation (3.100) takes the form

$$f(\mathbf{x}) = \operatorname{sgn}\left(R^2 - \sum_{i,j=1}^{m}\alpha_i\alpha_j(\mathbf{x}_i \cdot \mathbf{x}_j) + 2\sum_{i=1}^{m}\alpha_i(\mathbf{x} \cdot \mathbf{x}_i) - (\mathbf{x} \cdot \mathbf{x})\right). \tag{3.101}$$

The flexibility of the SVDD algorithm can be improved by noticing that the data appear only as dot products in Equation (3.101) and, therefore, the kernel trick can be used. SVDD and one-class SVM algorithms give similar results when using a Gaussian kernel (Schölkopf et al., 2001; Tax and Duin, 1999).

3.4 Concluding Remarks

In this chapter, statistical learning theory foundations were discussed in which the properties of learning algorithms and hypothesis space necessary for learning using the minimization of

training error or ERM criterion were highlighted. In particular, it was shown that minimizing the training error only using a finite sample was not sufficient for good generalization performance. It is also necessary to constrain the hypothesis space. The concept of the large margin was introduced that incorporates the learning bias for improved learning machine performance. Kernel functions were incorporated for (implicit) learning of linear decision functions in feature space corresponding to nonlinear decision functions in input space. Both supervised and unsupervised learning frameworks using the two ideas of margin maximization and kernel trick were introduced.

Chapter 4

Classification of Process Dynamics

To see a thing one has to comprehend it....If we really saw the world, maybe we would understand it.

Jorge Luis Borges, *There are more things*

METALLURGICAL and other chemical process systems are often too complex to model from first principles. In such situations the alternative is to identify the systems from historic process data. Such identification can pose problems of its own and before attempting to identify the system, it may be important to determine whether a particular model structure is justified by the data before building the model. For example, the analyst may wish to distinguish between nonlinear (deterministic) processes and linear (stochastic) processes to justify the use of a particular methodology for dealing with time series observations, or else it may also be important to distinguish between different stochastic models. In feedback controlled systems compensatory adjustment of manipulated variables is known to have a masking effect when certain faults occur in the process. This often results in the propagation of plant-wide oscillatory trends that may go unnoticed because of their multiscale nature. There is, therefore, a need for tools that can detect presence of such oscillatory trends so as to minimize their potentially negative impact on process quality and costs.

In this chapter the use of a linear method called singular spectrum analysis (SSA) for the classification of time series data is discussed. The method is based on principal component analysis of an augmented data set consisting of the original time series data and lagged copies of the data. In addition, a nonlinear extension of SSA based on kernel-based eigenvalue decomposition is proposed. The usefulness of kernel SSA as a complementary tool in the search for evidence of nonlinearity in data or for testing other hypotheses about such data is illustrated by simulated and real case studies.

4.1 Introduction

Reliable and effective process control is vital to the efficient operation of chemical process systems. The increasing emphasis on advanced (model-based) control systems in industrial applications requires a solid grasp of the dynamic behavior of the system or, failing that, at least some timely, reliable diagnostics of the process dynamics of the system. The search for evidence of predictability (or determinism) in observed data provides a starting point for system identification and design of advanced control systems. The extent of predictability, particularly for nonlinear systems, may be informative and hence the value of techniques designed for the detection of periodicities or intermittent trends in the data. For example, limit cycle oscillations which arise from faults within a feedback loop and subsequently propagated to other unit operations because of physical coupling and stream recycling are an important fault class in industrial problems (Thornhill, 2005). Likewise, closed loop identification of process systems requires knowledge of the nature of disturbance (stochastic models) affecting the process – information that may not be readily available. In this context, singular spectrum analysis (SSA) is a relatively new technique that can, among other, be used to test hypotheses about time series data or detection of oscillatory behavior. Initially developed in the field of climatology (Broomhead and King, 1986; Vautard and Ghil, 1989; Vautard et al., 1992), it has since been used in various research fields, including the biosciences (Mineva and Popivanov, 1996), geology (Rozynski et al., 2001; Schoellhamer, 2001), economics (Ormerod and Campbell, 1997) and solar physics (Kepenne, 1995). Essentially, SSA is a nonparametric approach capable of localizing intermittent modes in time and space. It is useful for identifying interesting dominant modes in observed data that are often missed by other spectral methods. The SSA technique involves sliding a window down a time series in order to identify patterns which account for a large proportion of the variance in these views of the time series. Monte Carlo singular spectrum analysis (MC-SSA) is a methodology for discriminating between various components of time series data, particularly between components containing meaningful information and other components containing mostly noise (Allen and Smith, 1996a,b). The technique also appeals to applications in process engineering, especially in model fitting for control and monitoring purposes, where observations on plants typically yield short time series corrupted with measurement and dynamic errors. A few process engineering applications of SSA and related techniques have been reported to date, for example, Aldrich and Barkhuizen (2003); Barkhuizen (2003); Thornhill (2005).

Unfortunately, SSA and MC-SSA only exploit linear correlations in data and could be of limited usefulness in the presence of non-trivial structures in the data that can only be described adequately by nonlinear relations. Nonlinear SSA extensions using multilayer perceptron (MLP) networks have been proposed and applied mainly in climatology studies (Hsieh, 2001, 2004; Hsieh and Wu, 2002a,b). In the following, an alternative nonlinear extension of SSA based on unsupervised kernel learning methods is introduced and compared with previously proposed approaches via simulation studies. Additionally, the proposed approach is extended to (nonlinear) MC-SSA in testing hypotheses of underlying process dynamics. As discussed later, when using MLP networks it is difficult to perform hypothesis testing except indirectly using, for example, model residual error analysis. However, model fitting is also prone to uncertainties whereas a direct test is possible with the proposed kernel-based approach.

4.2 Singular Spectrum Analysis

4.2.1 Background

Traditional spectral methods, for example spectral correlogram analysis, fit data to a pre-specified model by optimizing the weights over a fixed set of basis functions such as sine waves or wavelets models (Blackman and Tukey, 1958; Ghil et al., 2002). As is common with parametric approaches, large volumes of data with minimal noise are required to identify complex dynamical behavior that may be exhibited by the physical system of interest. In contrast, SSA methods use a data-adaptive basis set based on the information in the spectral coefficients of the data, thereby circumventing some of the limitations imposed by short and noisy time series routinely encountered in practice. Given an ordered sequence of observations, basic SSA decomposes the series into additive components that can be grouped, for example, into deterministic and stochastic processes. This split allows for many applications, such as model structural analysis, system identification, signal-to-noise ratio enhancement, and data compression.

The general SSA approach is based on the classical Karhunen-Loève (KL) orthogonal decomposition of a covariance matrix. Broomhead and King (1986) extended SSA to nonlinear dynamical systems theory using an orthogonal expansion of a trajectory matrix formed from lagged copies of a univariate time series. Vautard and Ghil (1989) formalized and exploited the duality between eigenspectral decomposition and the method of delays. In particular, they showed that the occurrence of a close symmetric-antisymmetric eigenvalue pair is associated with a nonlinear anharmonic oscillator. The nature of these oscillations is automatically determined from the data, which is not possible with classical spectral analysis, where the broad range of fixed basis functions can fail to localize intermittent oscillations.

4.2.2 Singular Spectrum Analysis Methodology

SSA involves four basic steps (Golyandina et al., 2001; Vautard et al., 1992) which are described in detail below.

Step I Time series embedding

Given a time series x_t, $t = 1, \ldots, n$, where n is the length of the time series, a trajectory matrix is constructed by sliding a window of length d along the time series to give lagged vectors $\mathbf{x}_i \in \mathbb{R}^d$:

$$\mathbf{x}_i = \begin{bmatrix} x_i & x_i + 1 & \ldots & x_{i+d-1} \end{bmatrix}', \text{ for } i = 1, 2, \ldots, n - d + 1. \tag{4.1}$$

The vectors $\mathbf{x}_i$ thus formed are collected in an augmented multidimensional time series referred to as the trajectory matrix:

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} x_1 & x_2 & \ldots & x_{1+d-1} \\ x_2 & x_3 & \ldots & x_{2+d-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-d+1} & x_{n-d+2} & \ldots & x_n \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_{n-d+1} \end{bmatrix}' \end{aligned} \tag{4.2}$$

Step II Singular value decomposition

A square covariance matrix $\mathbf{C}_X$ is evaluated from the trajectory matrix (Broomhead and King, 1986):

$$\mathbf{C}_X = \frac{1}{n-d+1}\mathbf{X}'\mathbf{X}.$$

Alternatively, assuming stationarity, one can impose a Toeplitz structure on $\mathbf{C}_X$ by enforcing constant diagonal entries (Vautard and Ghil, 1989). This has the effect of weighting the contribution of the end-points to the covariance matrix equally with the rest of the data[1]. The resulting covariance matrix is then given by

$$\mathbf{C}_X(i,j) = \frac{1}{n-|i-j|}\sum_{t=1}^{n-|i-j|} x_t x_{t-|i-j|}. \tag{4.3}$$

Differences in the formulations of $\mathbf{C}_X$ are generally significant only in the analysis of short time series (Allen and Smith, 1996b). Irrespective of the method used in estimating $\mathbf{C}_X$, an eigenvalue decomposition of the covariance matrix is obtained according to Equation (2.1), that is

$$\mathbf{C}_X\mathbf{p}_k = \lambda_k\mathbf{p}_k, \text{ for } k = 1,\ldots,d$$

where $\mathbf{p}_k$ and λ_k are the respective k^{th} eigenvector and eigenvalue. Each obtained eigenvector is sometimes referred to as an *empirical orthogonal function*. The square roots of the (non-negative) eigenvalues $\sqrt{\lambda_k}$ are called the singular values, and the set of ordered singular values $\sqrt{\lambda_1} > \sqrt{\lambda_2} \geq \ldots \geq \sqrt{\lambda_d} \geq 0$ is called the singular spectrum. The ordering implies that the k^{th} eigenvalue explains at least as much variance in the data compared to the $(k+1)^{\text{th}}$ eigenvalue.

Step III Grouping of Components

Projecting the embedded vectors $\mathbf{x}_i$ onto each principal direction $\mathbf{p}_k$ gives a time series $z_k(t)$ of length $(n-d+1)$,

$$z_k(t) = \sum_{j=1}^{d} x(t+j-1)\mathbf{p}_k(j), \text{ for } t = 1,2,\ldots,n-d+1. \tag{4.4}$$

The principal components or scores $z_k(t)$'s are representations of the original time series in the rotated coordinate space. Typically $q < d$ leading components are selected to explain the signal, effectively filtering high-frequency components in the data. The q-dimensional score vectors of the projected matrix Z are given by

$$\mathbf{z}(t) = [z_1(t)z_2(t)\ldots z_q(t)]', \text{ for } t = 1,2,\ldots,n-d+1. \tag{4.5}$$

[1] cf. Leakage at end-points in Fourier spectral analysis.

Step IV Time series reconstruction

Convolution of a set of principal components **Z** with the corresponding eigenvectors recovers phase information lost in the decomposition step;

$$\tilde{x}(t+j-1) = \sum_{k=1}^{q \leq d} z_k(t)\mathbf{p}_k(j), \tag{4.6}$$

for $t = 1, 2, \ldots, n-d+1$, and $j = 1, 2, \ldots, d$. Reconstruction of the time series can be performed via a diagonal averaging procedure (Golyandina et al., 2001):

$$\tilde{x}_i = \begin{cases} \frac{1}{i}\sum_{j=1}^{i}\sum_{k=1}^{q} z_k(i-1)\mathbf{p}_k(j), & \text{for } 1 \leq i \leq d-1 \\ \frac{1}{d}\sum_{j=1}^{d}\sum_{k=1}^{q} z_k(i-j)\mathbf{p}_k(j), & \text{for } d \leq i \leq n-d+1 \\ \frac{1}{n-i+1}\sum_{j=1-n-d}^{d}\sum_{k=1}^{q} z_k(i-j)\mathbf{p}_k(j), & \text{for } n-d+2 \leq i \leq n \end{cases} \tag{4.7}$$

The diagonal averaging (Step IV) is an adaptive optimal filtering method in the least squares sense that works even well for short data sets (cf. Wiener filter) (Vautard et al., 1992). The four steps are schematically illustrated in Figure 4.1.

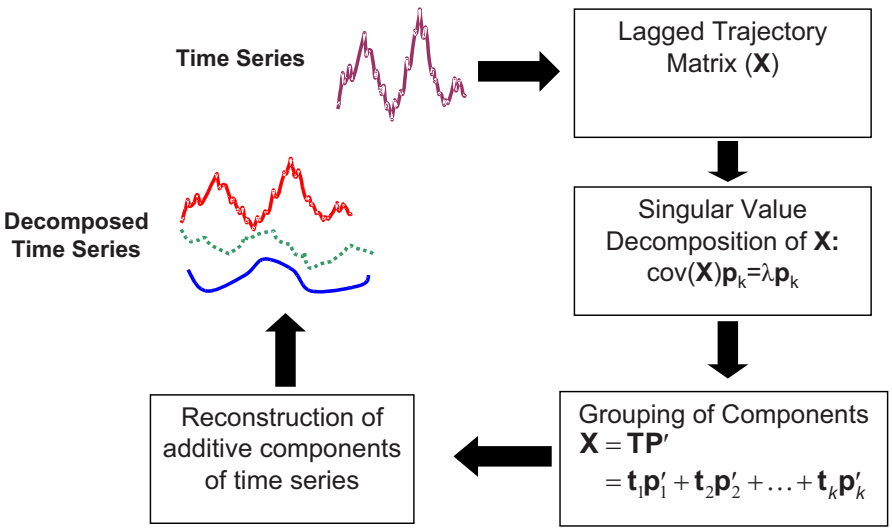

Figure 4.1: A schematic illustration of the SSA methodology

The SSA procedure as described can be carried out for any set of eigenvectors. Thus, an appropriate grouping of the eigenvalue indices can be performed to yield different components of the time series, such as trends, oscillations or noise. Also, for some purposes it may not be necessary to perform all the steps.

The SSA approach is closely related to the multivariate statistical technique of principal component analysis (PCA)(Jolliffe, 2002). In fact, SSA is PCA performed on the trajectory or lagged matrix, Equation (4.2). Hence, all mathematical and statistical properties associated with PCA apply to SSA. In particular, SSA performs a rotation of the coordinate space such that the first principal direction explains maximal variance compared to other directions; the principal components or scores are uncorrelated; the approximation error incurred in representing the multivariate data $\mathbf{X}$ by the first q principal components is minimal, and the first q components have minimal entropy with respect to the inputs, assuming the data have a Gaussian distribution.

4.2.3 Limitations of Singular Spectrum Analysis

Classical PCA is a linear multivariate statistical useful for extracting structure from multidimensional data. Unfortunately, being a linear technique, it is not guaranteed to capture subtle nonlinear or other complex structure that may be present in data. A typical intuitive example is in digit recognition using image analysis. The information characterizing a particular digit is concentrated in regions of the image as indicated by the pixels. It can therefore be expected that the reduced space containing enough descriptive and/or discriminative information is derived through higher order correlations between the individual pixels.

As highlighted in Section 3.3.1, there have been efforts to extend the otherwise powerful technique of PCA to handle nonlinear structures in data. With respect to specific nonlinear SSA extensions, auto-associative MLP-based methods that use a circular "bottleneck" layer have been reported (Hsieh, 2001, 2004; Hsieh and Wu, 2002a,b). However, as with other methods that use neural networks, the approach is prone to entrapment in local minima. Also, MLP-based methods are not amenable to direct hypothesis testing of time series structure, except through residual analysis.

In the following, a kernel-based nonlinear generalization of PCA first suggested in Schölkopf et al. (1998) is proposed as an alternative nonlinear SSA method. Unlike the MLP-based approach, nonlinear SSA using kernels is readily extendable to hypothesis testing of time series as done in Monte Carlo singular spectrum analysis (MC-SSA). Before discussing the nonlinear SSA method using kernels, first a brief overview of hypothesis testing for time series classification using SSA is in order.

4.2.4 Nonlinearity Testing Using Monte Carlo Singular Spectrum Analysis

The method of surrogate data is used to test for evidence of interesting structure in measured data in physical systems (Schreiber and Schmitz, 2000; Theiler and Prichard, 1996). The procedure initially suggests a null hypothesis H_0 about the system generating the data. For example, in nonlinear time series analysis a commonly used hypothesis is that the data follows a linear, Gaussian, stationary, stochastic dynamical rule. The alternative hypothesis H_1 would imply that the time series is nonlinear deterministic. An empirical quantity or test statistic T_0 that can be evaluated on the data and has high power to reject H_0 is then chosen according to the specific type of the alternative hypothesis. An example

of a robust statistic motivated by the search for evidence of chaotic phenomena in dynamic systems is the correlation dimension (Small and Judd, 1998).

Next, an ensemble of surrogate data x_t^{surr} consistent with H_0 are generated. Test statistics computed for both the original x_t and surrogate data are subsequently compared. If T_0 for x_t is significantly different from the expected T_0 under the null H_0 (given the data) as estimated according to x_t^{surr} the null is rejected, otherwise it is accepted. However, rejection of H_0 does not indicate positive evidence of the alternative. For such a conclusion to be made, it is necessary to use a test statistic that has high power to detect only deviations distinct from the alternative, and not other possible alternatives (Timmer, 2000). Clearly therefore, a crucial aspect of the procedure is the selection of appropriate test statistics.

As an example, to test whether a time series contains any structure, (that is, whether it is random "white" noise or not) surrogate data could be generated by randomizing the time series and comparing the estimated autocorrelation functions (i.e., the test statistic) of the original time series and its surrogates. Alternatively, models can be fitted to the randomized ensemble of surrogates and the original time series, with the prediction errors serving as test statistics. Lack of significant differences between the computed statistics would imply that the original time series is (most probably) a sample from the same distribution that generated the surrogates. In this case both the autocorrelation functions and prediction errors would be appropriate discriminating test statistics, while data averages, variances or other similar statistics not related to the structure of the time series data would not be. The use of surrogate analysis in process engineering has been reported in Barnard et al. (2001); Theron and Aldrich (2004); Thornhill (2005).

Monte Carlo SSA (MC-SSA) is a variation of the method of surrogate analysis described above. In the presence of pure white noise SSA is guaranteed to identify any modulated oscillatory modes present in the data. However, reliable identification of oscillatory modes is difficult when the noise process has first-order autocorrelation. Such so-called AR(1) processes exhibit large power at low frequencies but cannot support oscillations. These AR(1) processes are usually referred to as "red noise" in climatic time series analysis (Allen and Smith, 1996b; Ghil et al., 2002). A robust test, called the Monte Carlo SSA test, with statistically improved signal detection properties was proposed in Allen and Smith (1996b). The test compares the distribution of test statistic values obtained from simulated red noise processes with the corresponding value for the observed time series. Although applicable in different contexts, Monte Carlo SSA has largely been applied in distinguishing time series data from AR(1) or "red noise" processes of the form

$$x_t = \gamma(x_{t-1} - \bar{x}) + \alpha\varepsilon_t + \bar{x}, \tag{4.8}$$

where ε represents independent and identically distributed noise, $\bar{x}$ is the process mean, and (γ, α) are process parameters. Since the process mean can always be zeroed, only two noise parameters need to be estimated, viz. γ and α. The Best Linear Unbiased Estimators (BLUE) of the noise parameters can be obtained using generalized linear regression techniques, thus ensuring that the obtained estimates maximize the likelihood of accepting the entire class of AR(1) processes. In the simulation results reported here, BLUE estimates were used for the noise parameters based on corrected estimators of Allen and Smith (1996b).

MC-SSA proceeds by using the parametric model described by Equation (4.8) to generate an ensemble of surrogate data or realizations of a red noise process. Each surrogate time series is embedded into a trajectory matrix as in Equation (4.2). The covariance matrix of the surrogate set is evaluated and decomposed according to

$$\lambda_{k,\text{surr}} = \mathbf{p}_{k,\text{surr}}{}' \mathbf{C}_{\text{surr}} \mathbf{p}_{k,\text{surr}}, \text{ for } k = 1, \ldots, d. \tag{4.9}$$

By repeating the same computation for each surrogate, it is possible to define confidence bounds on the distribution of the eigenspectra obtained from the surrogates for a given significance level, usually $\alpha = 0.05$ (Dettinger et al., 1995a,b; Elsner and Tsonis, 1996).

In spectral analysis and related studies where the interest is in identifying oscillatory modes, it is usual to search for pairs of eigenvalues above a given threshold, say the 97.5^{th} percentile. An accurate interpretation requires further analysis, including two-way Monte Carlo pass (Allen and Smith, 1996a). However, comparing the overall shape of the eigenspectra of the data and surrogates is useful for the purposes of distinguishing signals from red noise. This approach will be referred to as the eigenshape or Elsner-Tsonis test.

An alternative approach proceeds by projecting each surrogate covariance matrix onto the original eigenbasis of the data in Equation (4.3) (Allen and Smith, 1996b);

$$\hat{\lambda}_k = \mathbf{p}_k{}' \mathbf{C}_{\text{surr}} \mathbf{p}_k. \tag{4.10}$$

Hereinafter, surrogate tests based on Equation (4.10) test will be referred to as Allen-Smith or projection onto data eigenbasis test.

Using statistical tests, the distributions of the surrogate eigenspectra are compared with that of the original data, from which a decision can be made either to reject the null hypothesis or to accept it, depending on the desired significance level(s). Allen and Smith (1996a,b) have also extended this standard test to the case where one or more signals have been detected and removed and the structure of the residuals needs to be examined.

The two eigenspectrum tests have their merits, depending on the application. The Elsner-Tsonis (or eigenshape) test (Equation 4.9) compares the overall shapes of the eigenspectra of the surrogates and the original data to determine whether the observed time series is different from red noise. If one is interested in the structure of the eigenvectors to detect oscillatory behavior, for example, this approach fails, since there is no unique k^{th} eigenvector for the surrogates.

The alternative method of projecting of surrogates onto the data eigenbasis (Equation 4.10) is useful for getting more information about the eigenvectors. A major criticism of this approach is that the projection of the surrogates onto data eigenvectors results in a matrix of singular values with non-zero off-diagonal elements. If these non-zero off-diagonal elements are within range of an order of magnitude to the diagonal elements, the interpretation of the results becomes unreliable (Elsner and Tsonis, 1996).

4.2.5 Nonlinear Singular Spectrum Analysis

As mentioned earlier, the SSA approach as discussed so far is restricted to exploiting information related to linear correlations in multivariate data to reveal interesting latent

structures. However, the tools may have limited applicability when significant nonlinear correlations exist in the data. Hsieh (2004) describes an extension of basic SSA to a nonlinear approach that uses auto-associative neural networks to exploit nonlinear redundancies that may exist in the data that are not detected by the basic approaches discussed above. In this particular contribution the nonlinear approach is extended by introducing kernel-based singular spectrum analysis. In the main, the focus will be on time series classification using these Monte Carlo innovations. However, the extension to other applications follows naturally. Below, MLP-based nonlinear SSA is briefly discussed.

Nonlinear SSA using Multilayer Perceptrons

Nonlinear spectral analysis based on multilayer perceptron (MLP) networks were proposed in Hsieh (2001, 2004). In this approach, a MLP network learns a forward mapping from the d-dimensional input space to the reduced space or bottleneck layer (one-dimensional) using an encoding layer of r neurons. An inverse mapping is achieved by mapping back the bottleneck layer outputs to the inputs via a decoding layer with r' neurons as shown in Figure 4.2. Such a network topology in which a set of inputs is mapped onto itself is known as auto-associative learning. For simplicity, r' is usually enforced to be equal to r. It is very similar to Kramer's auto-associative MLP method (Kramer, 1992), except for the bottleneck structure, which has two nodes restricted to lie in a unit circle to give a single degree of freedom or nonlinear principal component θ. Such an architecture (hereinafter called the NLPCA.cir following Hsieh (2001)) is able not only to model open curve solutions but closed solutions as well. It has the desirable capability to extract periodic or wave modes in data.

Although each of the encoding and decoding layers is not restricted to unit size, it is not generally necessary to use more layers as a single hidden layer with enough nodes can approximate any nonlinear function to an arbitrary accuracy (Kramer, 1992). The optimal number of nodes in the hidden layer (r) is determined by minimizing the sum squared error between the inputs and outputs. To avoid overfitting, the data are split into training and validation sets over which the optimal structure is determined. Unfortunately, as with other gradient descent optimization methods, NLPCA.cir is not guaranteed to converge to the global optimal solution. Also, training over different training/testing ensembles is computationally costly, and it is generally difficult to get the same optimal solution on re-training the same neural network (Bishop, 1995).

Nonlinear Monte Carlo SSA

Recall that the projection of the covariance matrices of the surrogate data onto the eigenbasis of the data gives an expected distribution of the singular spectrum of the null hypothesis, if it were true. For the kernel-based approach equivalent formulations are proposed, with the exception that the evaluations are in feature space. Hence, consider the diagonalization of the kernel matrix (see Section 3.3.1),

$$\lambda_k = \mathbf{a}_k \mathbf{K} \mathbf{a}_k, \text{ for } k = 1, \ldots, d. \tag{4.11}$$

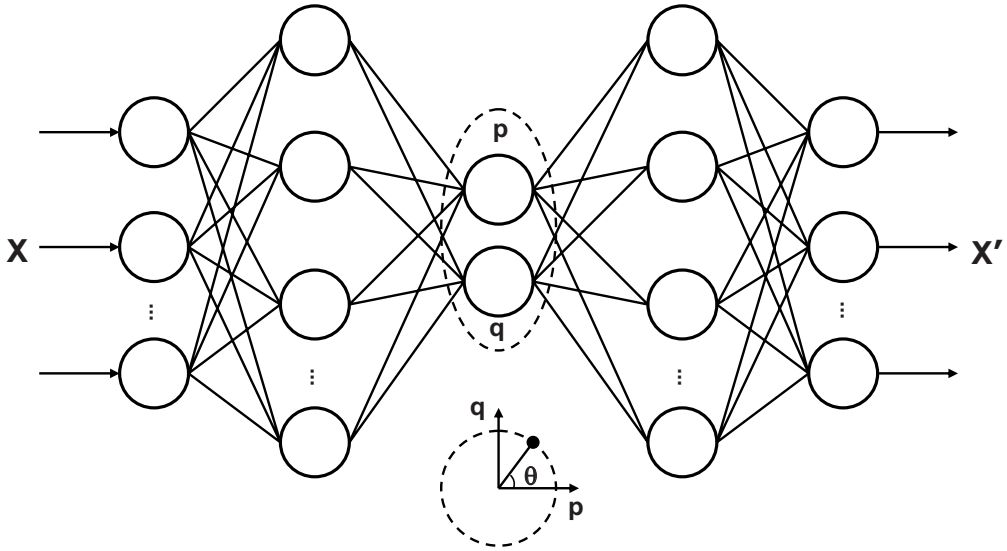

Figure 4.2: Architecture of the NLPCA.cir multilayer perceptron model with a circular bottleneck node. The different layers are referred to as (left to right) input layer (d nodes), encoding layer (m nodes), bottleneck layer (2 nodes), decoding layer (r nodes), and the output layer (d nodes). The model implements a forward nonlinear mapping from the input layer to the bottleneck layer, and an inverse mapping from the bottleneck layer to the output layer. The p and q nodes are confined to lie in a unit circle, thus representing a single degree of freedom θ (or nonlinear principal component). Not shown in the illustration are the bias nodes for each layer which, in practice, are included to allow for offsets.

The kernel equivalent formulation for the eigenshape test proceeds by evaluating the eigenvalues of the trajectory matrices of surrogate data in the feature space $\mathcal{H}$ induced by the chosen kernel function and comparing the resulting distribution to the eigenvalues of the image of the data in the same feature space. Similarly, the feature space equivalent of the Allen-Smith test projects the transformed surrogate data (in practice, the Gram matrix of the surrogate, $\mathbf{K}_{\text{surr}}$) onto the eigenbasis of the observed data in the feature space, that is

$$\hat{\Lambda} = \mathbf{A}'\mathbf{K}_{\text{surr}}\mathbf{A}. \tag{4.12}$$

The motivation for this approach lies in the capacity of kernels to extract nonlinear correlations when they are present in the data as will be illustrated in the next section. For example, nonlinear processes can be distinguished from linear stochastic processes, that is by testing the null hypothesis that the process is linear, against the alternative hypothesis that it is nonlinear. Similarly, the idea can be extended to any null hypothesis.

4.2.6 Case Study: Simulated Anharmonic Wave

To illustrate the capabilities of nonlinear SSA using kernel algorithms, consider the anharmonic wave simulation problem used by Hsieh and Wu (2002b). The anharmonic wave is

generated according to

$$g_t = \begin{cases} 3, & \text{for } t = 1, \ldots, 7 \\ -1 & \text{for } t = 8, \ldots, 28 \\ & \text{periodic thereafter.} \end{cases} \tag{4.13}$$

The stretched square wave for 600 samples is shown in Figure 4.3(a). Gaussian noise with twice the standard deviation of the stretched square wave was added to generate the noisy time series to be analyzed, as indicated in Figure 4.3(b). Following Hsieh and Wu (2002b), the leading 8 principal components obtained after performing linear SSA with a sliding window of length $d = 50$ were extracted and used as inputs for nonlinear SSA.

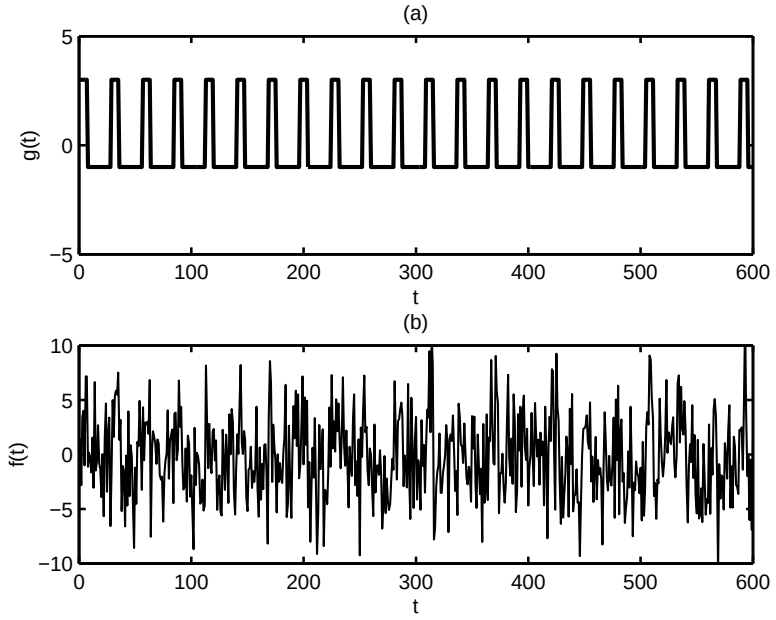

Figure 4.3: Stretched (anharmonic) square wave (a) without noise and, (b) with Gaussian noise added with zero mean and standard deviation twice that of the pure signal in (a)

As shown in Figure 4.4(a), Hsieh's NLPCA.cir network (using optimally determined m=8 nodes in the hidden encoding layer) is able to identify the closed curve related to the anharmonic wave. Nonlinear relations are also apparent between the first SSA linear mode and higher modes, Figures 4.44(b)–(d). Note that the results obtained here although comparable are different from those reported in Hsieh and Wu (2002b) where the optimal network structure used had $m = 5$ nodes. Despite repeated attempts aimed at optimizing the free parameter, their exact results could not reproduced, most probably owing to the presence of local minima encountered during optimization of the network.

The kernel PCA approach is also able to identify the closed curve solution as illustrated in Figure 4.5(a). Also, the nonlinear relations between the different linear SSA modes are clearly visible using the kernel approach compared to the MLP-based method in the previous figure, viz. Figures 4.4(b)–(d) (cf. Figures 4.4(b)–(d)). Moreover, the results are

stable across repeated simulations. This is hardly surprising as only linear algebra techniques are used to solve the eigenvalue problem in kernel PCA, with nonlinear optimization only performed in the preimage reconstruction.

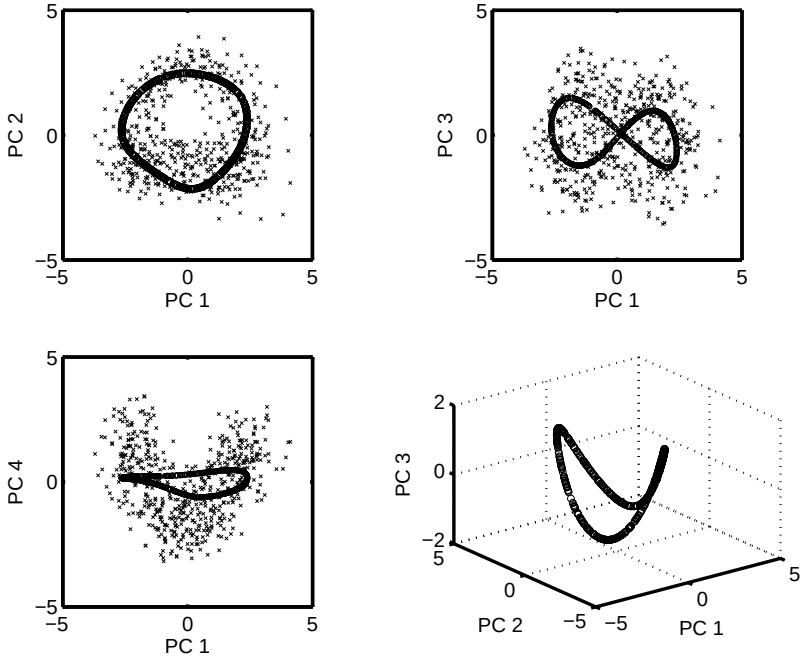

Figure 4.4: Projection of the NLSSA mode 1 onto the linear planes given by PC1-PC2 (top left); PC1-PC3 (top right); PC1-PC4 (bottom left) and PC1-PC2-PC3 (bottom right) principal directions. Note the poor estimate of the parabolic structure in the bottom left subplot, most probably as a result of the optimization getting trapped in local minima.

An important step in the SSA methodology is grouping of components to, for example, separate signal and noise in observed data. Similar to linear SSA, reconstructed components can be obtained using nonlinear methods. In the anharmonic wave example, the reconstructed components are obtained by projecting the nonlinear scores onto the corresponding nonlinear SSA principal directions to give an augmented matrix. Diagonal averaging gives the reconstructed time series using Equation (4.7).

Figure 4.6(a)–(b) show the linear reconstructed time series using the leading principal component (RC1-1) and the first 3 principal components (RC1-3) in the reconstructions respectively. Figures 4.7(a) and (b) show the reconstructed time series using NLPCA.cir mode 1 and kernel PCA using a Gaussian kernel of width $\sigma = 1.516$ respectively. For comparative purposes, a superimposition of the underlying stretched anharmonic wave is also shown on the plots. It can be seen that reconstructions based on the nonlinear approaches approximate the underlying anharmonic wave better than the linear methods. The kernel-based method, in particular, gives the best result overall as shown by its correlation coefficient with the square wave, as well as the variance explained in the noisy data, Table 4.1. Since the underlying square signal accounts for 22.6% variance in the noisy signal, linear reconstructions using 4 or more PCs explaining more than 22.6% variance include a

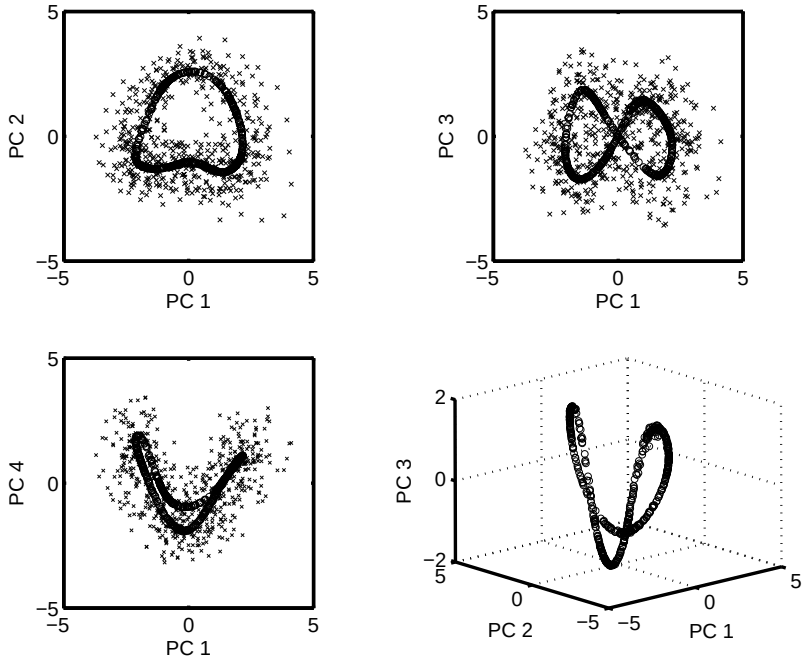

Figure 4.5: Projection of the kernel nonlinear principal components onto the linear planes given by (top left) PC1-PC2 (top right) PC1-PC3 (bottom left) PC1-PC4 and (bottom right) PC1-PC2-PC3 principal directions. A Gaussian kernel (width = 1.516) was used and four feature space modes were retained in the reconstruction.

large proportion of noise in the reconstruction. In other words, higher linear components have a low signal to noise ratio and, therefore, do not improve the reconstruction. Using all linear principal recovers the original noisy signal shown in Figure 4.3(b).

In summary, with proper kernel selection nonlinear SSA using kernel methods performs best in detecting the strongly anharmonic characteristics embedded in a noisy signal. Although linear SSA performs better than classical Fourier spectral energy analysis in detecting anharmonic waves (Hsieh and Wu, 2002b), nonlinear methods are best suited to recover signals exhibiting nonlinear properties.

4.3 Monte Carlo SSA Using Kernel PCA

4.3.1 Benchmark Systems

To assess the usefulness of the kernel-based MC-SSA approach, benchmark time series were generated as shown in Table 4.2. Each time series consisted of 200 data points. For each test a set of 1000 surrogate data sets were generated and the following tests performed:

- Eigenspectrum shape (Elsner-Tsonis) test using standard MC-SSA;
- Allen-Smith test using the data eigenbasis for projecting surrogate trajectory matrices;

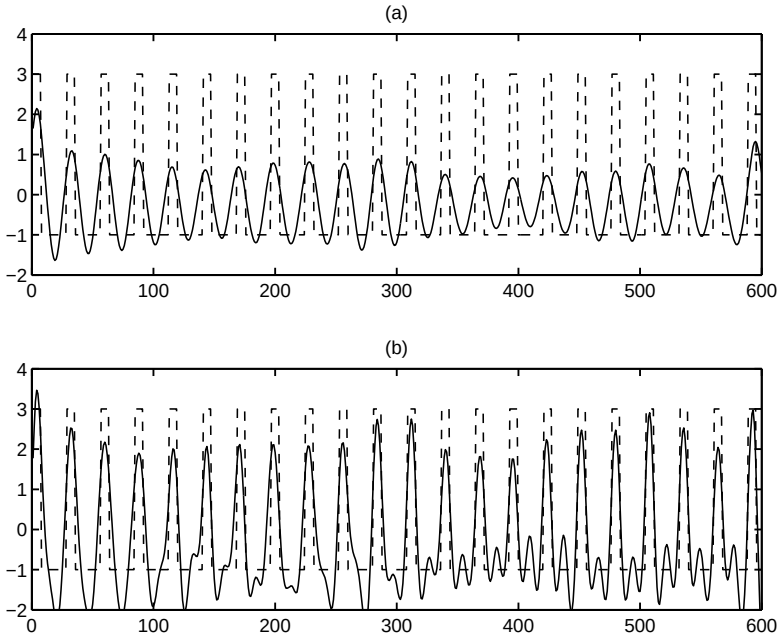

Figure 4.6: Reconstructed time series for the noisy square wave using the (a) leading linear principal component and (b) the first three linear principal components. The uncorrupted stretched square wave is shown superimposed (dashed line). Increasing the number of principal components in the reconstruction improves the approximation up to a point (3 principal components in this case).

- Eigenspectrum shape (or Elsner-Tsonis) test using kernel-based MC-SSA; and,
- Allen-Smith test using the data eigenbasis in kernel feature space for projecting surrogate trajectory matrices.

In all instances the null hypothesis stated that the time series had originated from an AR(1) process. Although any class of linear stochastic processes can be used as the null hypothesis, the AR(1) model was chosen as a null hypothesis, since AR(1) processes have no preferred frequencies (and thus oscillations can be identified). Moreover, time series modelling generally requires attributing at least some of the variability in observed time series to a stochastic residual term (Allen and Smith, 1996b), since measured or observed data are invariably corrupted with noise.

Gaussian and polynomial kernels were used. To check the influence of hyperparameters, the Gaussian width and polynomial degree were respectively varied as $\sigma = [0.01, 0.1, 1, 10, 100]$ and $\deg = [3, 5, 7, 9, 15]$. Additionally, in the case of polynomial kernel the Gram matrix was normalized for numerical reasons by requiring unity matrix diagonal entries, that is $\mathbf{K}_{i,j} = \mathbf{K}_{i,j}/\sqrt{\mathbf{K}_{i,i}\mathbf{K}_{j,j}}$.

4.3.2 Simulation Results and Discussion

The time series classification results using both linear and nonlinear (kernel-based) MC-SSA are presented in Tables 4.3 and 4.4 for eigenspectrum shape and projection onto data

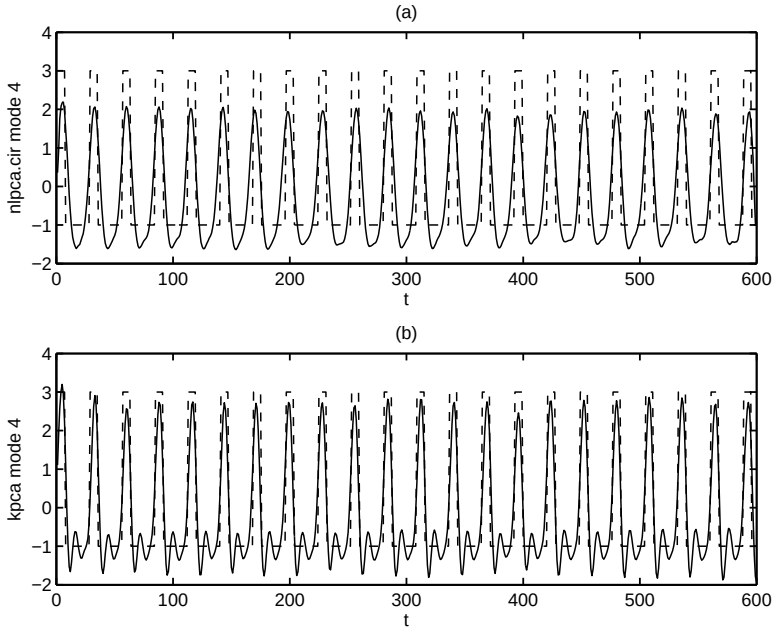

Figure 4.7: Reconstructed time series using nonlinear principal components based on (a) Hsieh's NLPCA.cir approach where only mode 1 is used (see text for details) and (b) kernel PCA using leading four features or "modes" from the kernel eigen-decomposition. Note the improved approximation of the underlying square wave obtained with kernel PCA (see also Table 4.1).

eigenbasis tests respectively. In Figures 4.8 and 4.9 are the corresponding plots of the tests. In the nonlinear case, results are only plotted for kernel parameters that gave the best performance. As mentioned previously, all results obtained are with reference to linear stochastic AR(1) surrogate realizations, that is

$$\begin{aligned} H_0 &: \ x(t) \text{ is consistent with an AR(1) process;} \\ H_1 &: \ x(t) \text{ is NOT consistent with an AR(1) process.} \end{aligned} \tag{4.14}$$

In general, the nonlinear tests tend to perform relatively better than the linear versions. However, the performance is sensitive to the choice of the kernel hyper-parameters. For example, in the case of the eigenshape test Table 4.3, the Gaussian kernel performance is drastically worse than the linear SSA for kernel width less than 1. On the other hand, for higher polynomial degree (> 7) the Type I error increased. A similar pattern is observed for the Gaussian kernel in the Allen-Smith (AS) test Table 4.4. However, a reversal of the pattern occurs in the case of the polynomial kernel where the Type I error is practically eliminated for polynomial degrees of degree 5 and above under the AS test.

It also interesting to note the shape of the eigenspectra in both tests. The surrogates' eigenshape closely resemble the eigenshape of the observed time series when using the eigenshape or Elsner-Tsonis test in both the linear and nonlinear variants. The opposite occurs for the Allen-Smith tests where the covariance matrices of the surrogates are projected onto the principal directions of the decomposed data covariance matrix. This results in a spread of the energy across off-diagonal elements. Since most of the energy is still

Table 4.1: Estimation accuracy of the underlying square wave from the noisy signal using 1–8 linear SSA components (RC1-1 – RC1-8) and nonlinear SSA with auto-associative MLP (MLP-NLRC1) and kernel PCA (kernel NLRC1)

	Correlation coefficient with "true" wave	Proportion of variance explained in noisy signal
TRUE square wave	1.000	0.226
RC1-1	0.698	0.117
RC1-2	0.755	0.145
RC1-3	0.849	0.214
RC1-4	0.849	0.236
RC1-5	0.828	0.290
RC1-6	0.770	0.312
RC1-7	0.771	0.361
RC1-8	0.750	0.386
MLP-NLRC1 (NLPCA.cir mode 1)	0.829(0.875*)	0.159(0.179*)
Kernel NLRC1 (#features=4)	0.907	0.213

* Results reported in Hsieh and Wu (2002b)

concentrated along the diagonal, the off-diagonal elements do not generally contain useful information. Unfortunately, especially in the nonlinear case where the Gram matrix has the same size as the number of data points, the spread of the energy across more dimensions results in eigenspectra different from the data. Hence, for reliable interpretation of the results under this test, it may be necessary only to consider a top fraction of the eigenspectra.

Based on these results, an improved strategy for testing unknown time series data can be formulated: when using the eigenbasis projection tests, if testing for the null hypothesis that the data are first order autoregressive, i.e. $H_0 : x(t) \sim x(t) = \alpha x(t-1) + \varepsilon(t)$, then accept the null hypothesis only if both the linear and the kernel methods (with hyperparameters set in the ranges indicated above) indicate acceptance of the hypothesis. This strategy is subsequently applied to case studies from the metallurgical industry.

Table 4.2: Benchmark systems

Model Description	**Mathematical Description**
First-order autoregressive process, ARX(t)	$x(t) = 0.92x(t-1) + \varepsilon(t), \varepsilon(t) \sim \mathcal{N}(0, 0.15)$
Autoregressive moving average ARMA(3,2), LGX(t)	$x(t) = \begin{cases} 0.12x(t-1) + 0.08x(t-2) - 0.2x(t-3) + \ldots \\ \varepsilon(t) + 0.15\varepsilon(t-1) + 0.36\varepsilon(t-2), \varepsilon(t) \sim \mathcal{N}(0, 0.15) \end{cases}$
Bilinear time series, BGX(t)	$x(t) = \begin{cases} 0.12x(t-1) + 0.08x(t-2) - 0.2x(t-3) + \ldots \\ \varepsilon(t) + 0.15\varepsilon(t-1) + 0.36\varepsilon(t-2) + \ldots \\ 0.40x(t-1)\varepsilon(t-1) + 0.16x(t-1)\varepsilon(t-2) - \ldots \\ 0.35x(t-2)\varepsilon(t-2), \varepsilon(t) \sim \mathcal{N}(0, 0.15). \end{cases}$
Nonlinear threshold autoregressive process, TGX(t)	$x(t) = \begin{cases} 0.1x(t-1) + \varepsilon(t), \text{ if } x(t-1) < 0.5 \\ 0.9x(t-1) + \varepsilon(t), \text{ if } x(t-1) \geq 0.5 \\ \varepsilon(t) \sim \mathcal{N}(0, 0.15) \end{cases}$
ARMA(3,2) linear process, LUX(t)	$x(t) = \begin{cases} 0.12x(t-1) + 0.08x(t-2) - 0.2x(t-3) + \ldots \\ u(t) + 0.15u(t-1) + 0.36u(t-2), u(t) \sim \mathcal{U}(-0.5, 0.5) \end{cases}$
Chaotic autocatalytic reactor (Lynch, 1992)	$\frac{\mathrm{d}x}{\mathrm{d}\tau} = 1 - x - Axz^2,$ $\frac{\mathrm{d}y}{\mathrm{d}\tau} = 1 - y - Byz^2,$ $\frac{\mathrm{d}z}{\mathrm{d}\tau} = 1 - (1+C)z + gAxz^2 + fByz^2$ where x, y, z are dimensionless state variables τ is dimensionless time, and $A = 18000, B = 400, C = 80, f = 4.2, g = 1.5$
Deterministic system, NLX(t)	$x(t) = \sin(t) + \cos\left(\frac{t}{2}\right) + \sin\left(\frac{t}{4}\right) + \cos\left(\frac{t}{8}\right), t \in [0, 12\pi]$

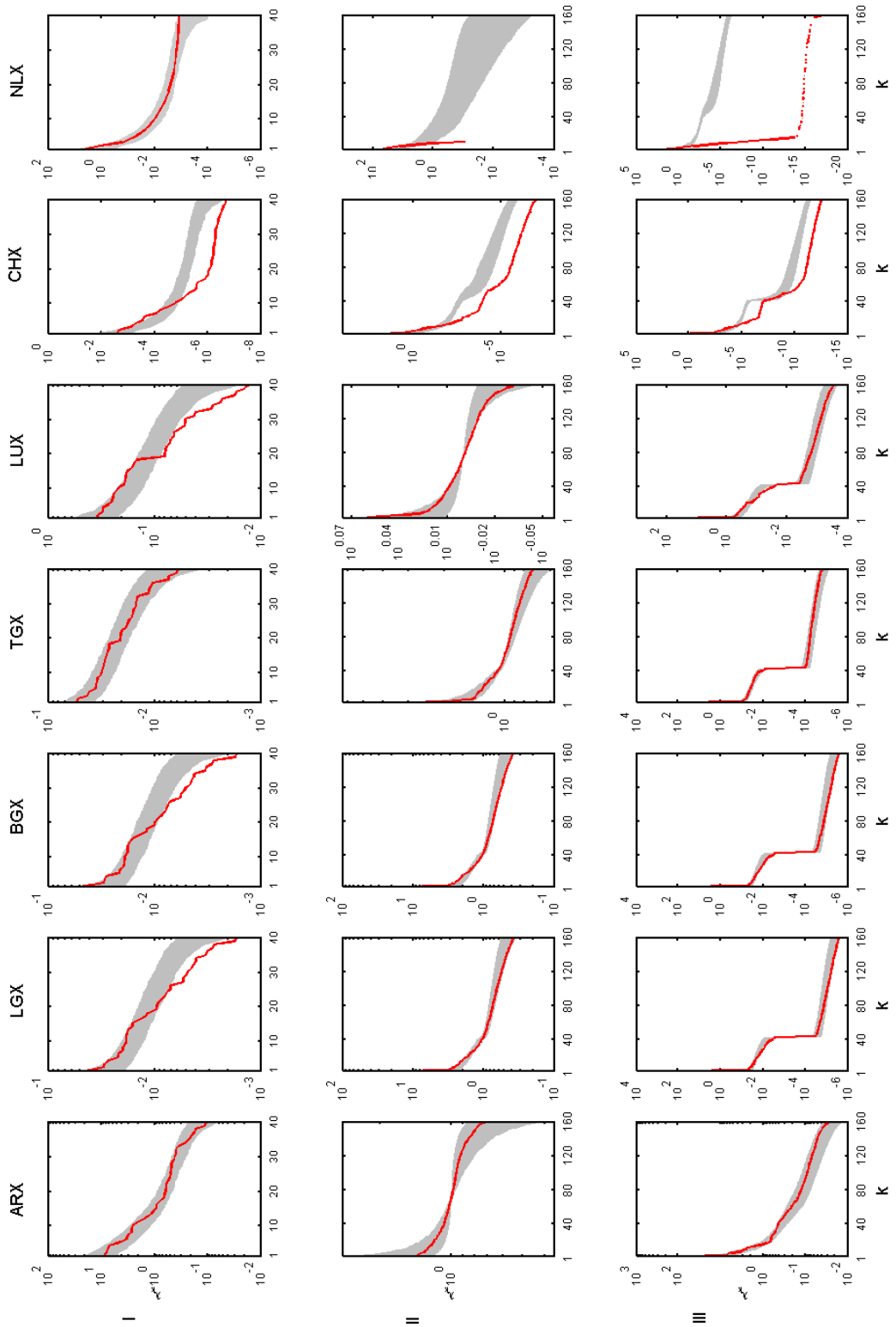

Figure 4.8: Elsner-Tsonis test results for the indicated time series using (I) linear (II) polynomial kernel with degree 7 and (III) radial basis kernel with width = 10. The shaded area represent the 95% confidence region generated from AR(1) process with parameters estimated using the data.

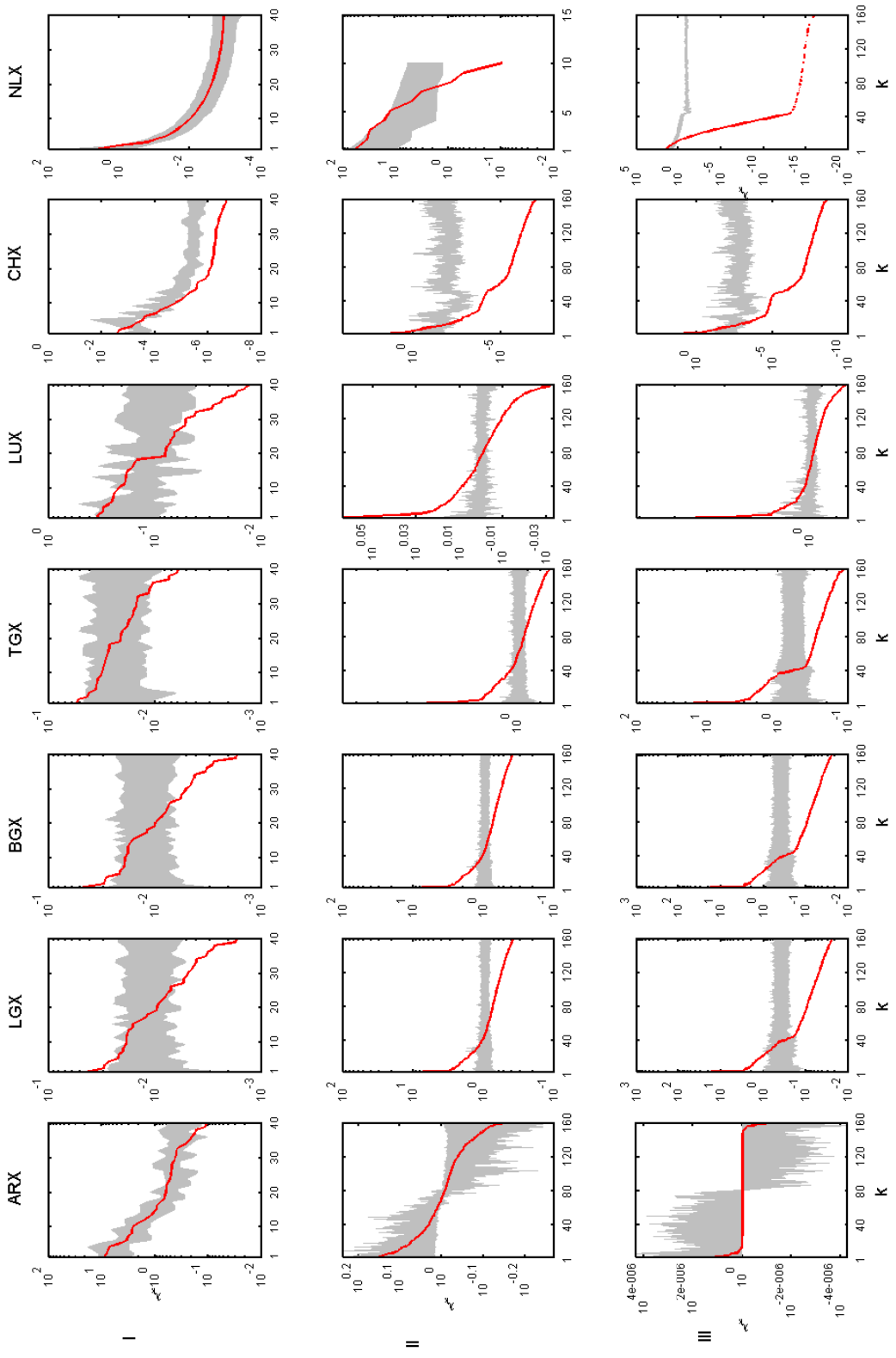

Figure 4.9: Allen-Smith test results for the indicated time series using (I) linear (II) polynomial kernel with degree 7 and (III) radial basis kernel with width = 1. The shaded area represent the 95% confidence region generated from AR(1) process with parameters estimated using the data.

Table 4.3: Performance of time series classification tests using standard and kernel-based FMU method. In the latter case, results are shown for different values of respective kernel hyperparameters.

Time Series	Linear	Gaussian kernel width					Polynomial kernel degree				
		$\sigma = 0.01$	$\sigma = 0.1$	$\sigma = 1$	$\sigma = 10$	$\sigma = 100$	deg $= 3$	deg $= 5$	deg $= 7$	deg $= 9$	deg $= 15$
ARX(t)	1 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)
LGX(t)	1 (0.47)	0 (0.00)	0 (0.00)	1 (0.11)	1 (0.11)	1 (0.11)	1 (0.12)	1 (0.09)	1 (0.05)	0 (0.00)	0 (0.00)
TGX(t)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.01)	0 (0.00)	0 (0.00)	0 (0.02)	0 (0.00)
BGX(t)	1 (0.50)	0 (0.00)	0 (0.00)	1 (0.14)	1 (0.12)	1 (0.12)	1 (0.23)	1 (0.12)	1 (0.06)	1 (0.01)	1 (0.01)
LUX(t)	1 (0.60)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.15)	1 (0.14)	1 (0.08)	1 (0.32)	1 (0.34)	1 (0.43)	1 (0.25)
CHX(t)	1 (0.82)	1 (0.97)	1 (0.90)	1 (0.94)	1 (0.92)	1 (0.25)	1 (0.97)	1 (0.97)	1 (0.97)	1 (0.97)	1 (0.96)
NLX(t)	0 (0.03)	1 (0.26)	1 (0.99)	1 (0.96)	1 (0.98)	1 (0.98)	1 (0.70)	1 (0.60)	1 (0.70)	1 (0.70)	1 (0.70)
	0.71	0.43	0.43	0.71	0.86	0.86	0.86	0.86	0.86	0.57	0.57

Table 4.4: Performance of time series classification tests using standard and kernel-based Allen-Smith method. In the latter case, results are shown for different values of respective kernel hyperparameters.

Time Series	Linear	Gaussian kernel width					Polynomial kernel degree				
		$\sigma = 0.01$	$\sigma = 0.1$	$\sigma = 1$	$\sigma = 10$	$\sigma = 100$	deg $= 3$	deg $= 5$	deg $= 7$	deg $= 9$	deg $= 15$
ARX(t)	1 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)	0 (0.88)	0 (0.96)	0 (0.44)	1 (0.09)	1 (0.00)	1 (0.01)	1 (0.00)
LGX(t)	1 (0.47)	0 (0.00)	0 (0.00)	1 (0.88)	1 (0.89)	1 (0.89)	1 (0.88)	1 (0.80)	1 (0.76)	1 (0.78)	1 (0.77)
TGX(t)	1 (0.20)	0 (0.00)	0 (0.00)	1 (0.91)	1 (0.94)	1 (0.94)	1 (0.86)	1 (0.76)	1 (0.73)	1 (0.70)	1 (0.67)
BGX(t)	1 (0.47)	0 (0.00)	0 (0.00)	1 (0.88)	1 (0.89)	1 (0.89)	1 (0.88)	1 (0.80)	1 (0.78)	1 (0.81)	1 (0.79)
LUX(t)	0 (0.50)	0 (0.00)	0 (0.00)	1 (0.66)	1 (0.90)	1 (0.94)	1 (0.83)	1 (0.85)	1 (0.83)	1 (0.78)	1 (0.59)
CHX(t)	1 (0.85)	1 (0.84)	1 (0.98)	1 (0.95)	1 (0.95)	1 (0.95)	1 (0.96)	1 (0.96)	1 (0.96)	1 (0.96)	1 (0.95)
NLX(t)	0 (0.00)	1 (0.26)	1 (0.99)	1 (0.94)	1 (0.98)	1 (0.98)	1 (0.50)	1 (0.40)	1 (0.40)	1 (0.40)	1 (0.50)
	0.86	0.43	0.43	1.00	0.86	0.86	0.86	0.86	1.00	1.00	1.00

4.3.3 Applications of MC-SSA on Metallurgical Plant Data

As with many chemical plants, metallurgical plants exhibit complex dynamical behavior arising from the interactions between unit reactors, feedback control, and also the underlying reaction chemistry. However, it is generally difficult to design nonlinear models with the capacity required for proper process monitoring and control. These systems are typically represented by linear models, which may not be optimal, or which may discount any deterministic components in the data *a priori*. These assumptions will be assessed by means of the techniques discussed in the preceding sections using two case studies drawn from real-world operating plants.

Case Study I: Recovery of Base Metals on a Flotation Plant

Background The data in this case study were obtained from a South African copper flotation plant. The plant consists of a crushing section and milling circuit, followed by a magnetic separation circuit. The purpose of the magnetic separation is to remove the high percentage of magnetic material in the ore and thereby reduce the load on the flotation circuit. The flotation circuit itself is designed to operate with feed grades of 0.6 % copper (Cu), 9.0 % lead (Pb), 2.4 % (Zn) and 130 g/t silver (Ag). The time series investigated were the measurements of the recovery grades of the precious metals, Cu and Pb in the scavenger circuit. Figure 4.10 shows the 12-minute interval measurements of the Pb and Cu concentrations. Each time series consisted of 1234 measurements and was pre-processed by scaling to zero mean and unit variance.

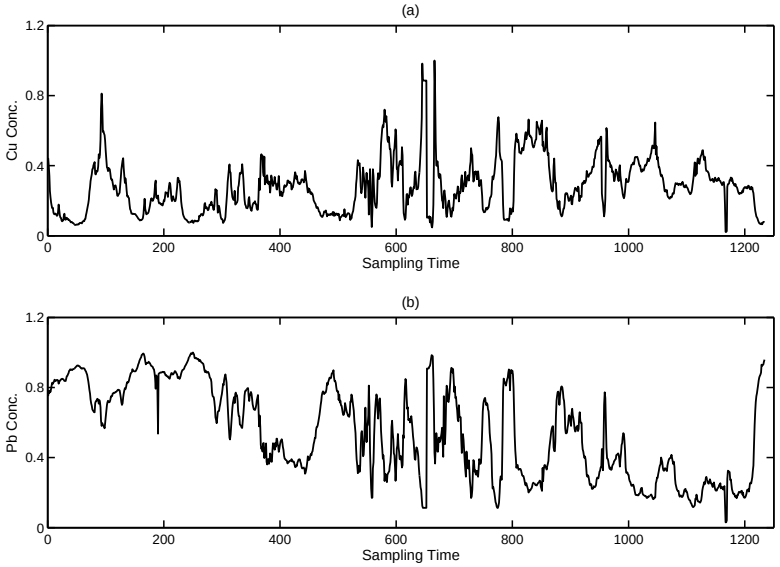

Figure 4.10: Time series plot of the variation of (scaled) concentration of (a) copper and (b) lead values in the scavenger stream of a milling circuit. The variables were sampled at 12-min intervals

Classification of the system The time series were tested against the hypothesis that the data had been generated by a first order autoregressive process and the eigenspectrum of the series was used as the test statistic. Linear and nonlinear eigenspectra obtained from PCA and kernel PCA decomposition of the lagged trajectory matrix of the time series were considered. A total of 15 first-order autoregressive surrogate series were used to generate 95% confidence limits for the eigenspectra of the series, which are displayed in Figures 4.11(a)–(d). Both of the tests indicated rejection of a first-order autoregressive process. Plots of the first three principal component score vectors of the lagged trajectory matrix of each time series (Figure 4.12) support these conclusions in that the attractors have a relatively smooth appearance, suggesting relatively small noise components in the data.

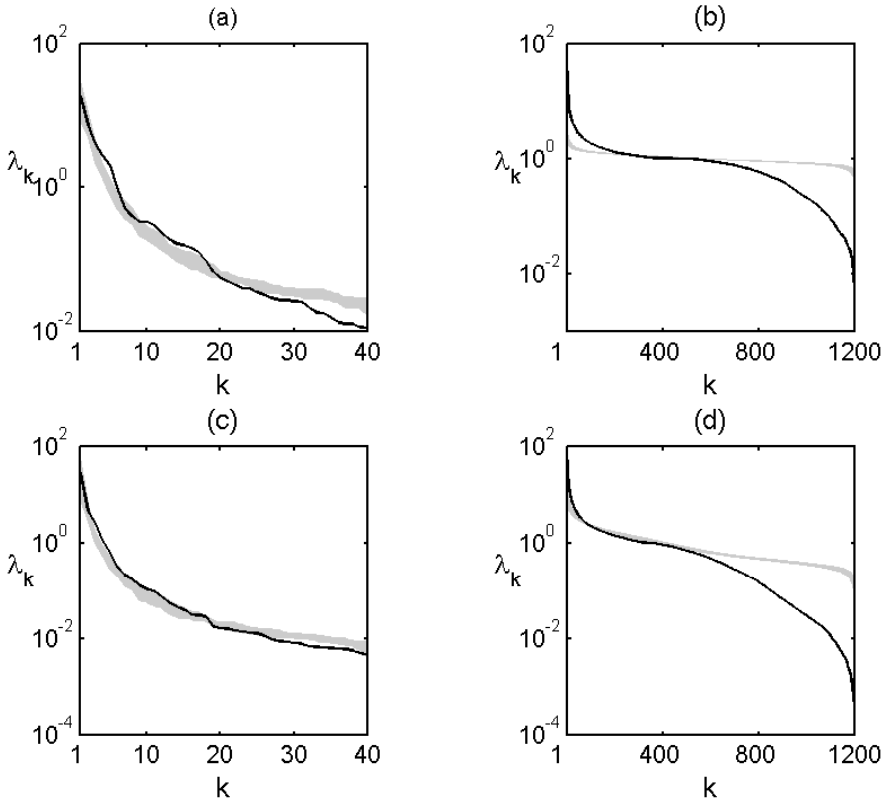

Figure 4.11: (a) Linear and (b) nonlinear (kernel) MC-SSA for the copper (Cu) time series; (c) linear and (d) nonlinear MC-SSA for the lead (Pb) time series. The shaded area represent the 95% confidence region generated from AR(1) process with parameters estimated using the data. Linear SSA rejects the null hypothesis in both cases although the split is more evident in the nonlinear case, where a Gaussian kernel (width $\sigma = 1$) was used.

Case Study II: Control of Acid Concentration in a Leach Circuit

The leaching of a valuable metal on an industrial plant is controlled by addition of acid to a series of leaching tanks. Manual dosage of the acid by an operator is complicated by the large residence time of the ore in the vessels, so that the effects of both over- and under-dosage are only discovered after the fact. A better understanding of the dynamics of the

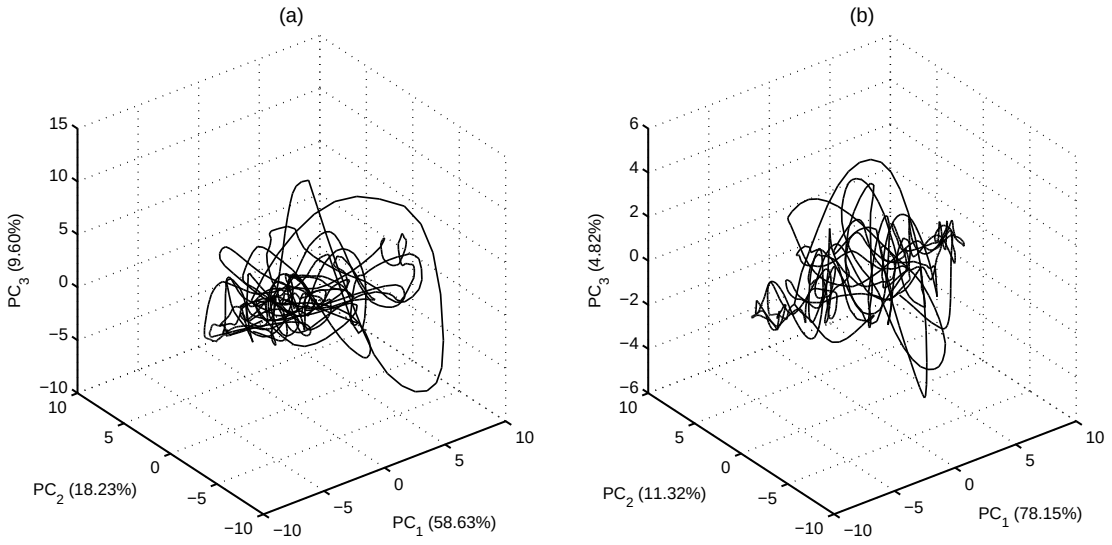

Figure 4.12: Attractors of (a) copper and (b) lead recoveries in the scavenger cell of a base metal flotation plant. The percentage of the total variance explained by each principal component is shown in parentheses in the appropriate axis label.

metal and the acid concentration could therefore lead to large improvement in the control of the leaching process by means of, for example, a model-based decision support system. The data in Figures 4.13(a) and (b) show the normalized concentrations of the H_2SO_4 in the feed and anolyte respectively. A total of 2282 twice-daily samples were considered.

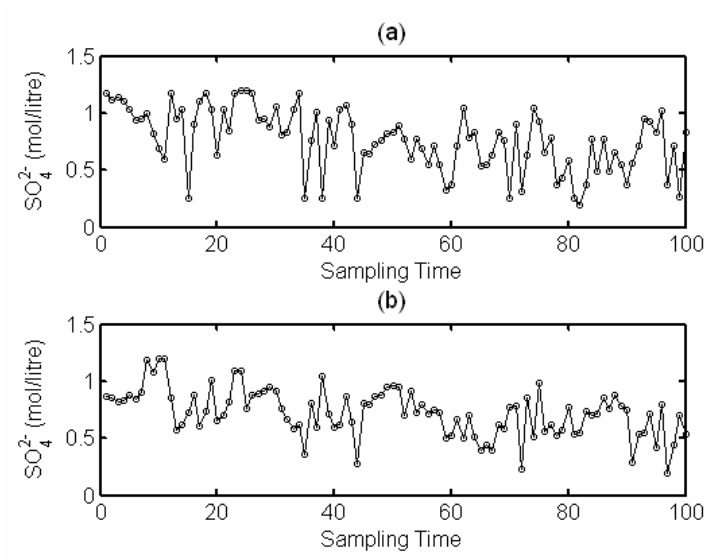

Figure 4.13: A plot of selected twice-daily observations of scaled sulphuric acid concentration in the anolyte (solid line) and feed (broken line) of an industrial leach plant.

Figures 4.14(a)–(d) shows the results of the Monte Carlo simulations under

H_0: $x(t)$ follows an AR(1) process.

In both cases the null hypothesis is rejected. The results are confirmed by a rudimentary linear autoregressive model fit to the data, the results of which are shown in Figure 4.15. In this figure, the broken line shows that the model order for the acid in the anolyte is approximately 3 or 4, while that of the acid in the feed is approximately 2 or 3. Although the linear fits were rudimentary (indicated by the very large errors), the purpose was to observe evidence of dynamic trends in the data. In this case, even the worst case analysis fails to account for the determinism which would have been expected if the null was true.

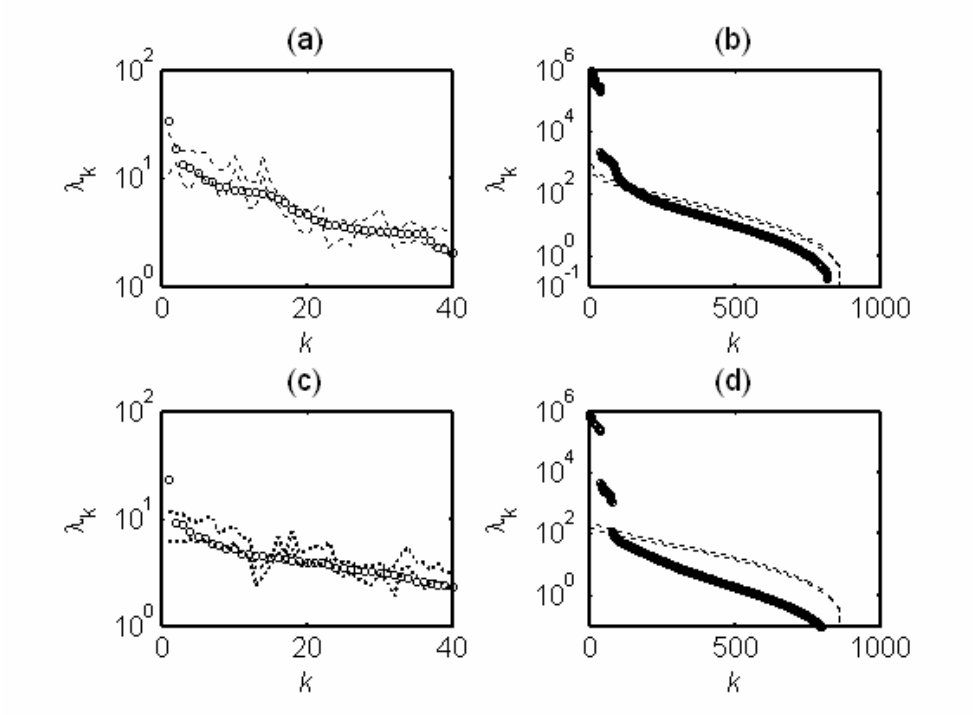

Figure 4.14: (a) Linear and (b) nonlinear MC-SSA hypothesis testing for the feed stream to the metal leaching circuit; (c) linear and (d) nonlinear MC-SSA hypothesis testing for the anolyte stream. The dashed lines indicate the estimated 95% confidence limits of the surrogate eigenspectra. A third-degree polynomial kernel was used in the nonlinear case.

4.4 Concluding Remarks

The basic formalism of SSA provides a natural test for periodic components in time series data against arbitrary stochastic models, such as the first order autoregressive models considered in this chapter (null hypotheses). At present application of the technique has been limited mostly to data related to the fields of climatology and meteorology, where different variants of the technique have been proposed. Although the use of these tests appears to have grown steadily over the last decade or so, the behavior of the tests is still not well understood.

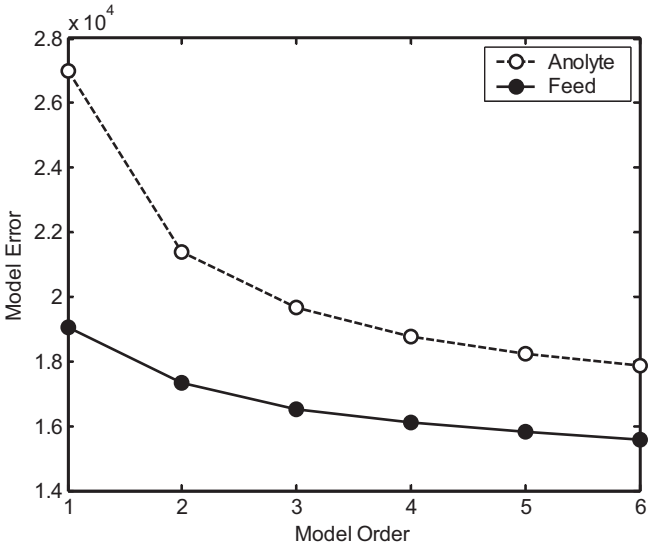

Figure 4.15: Predictive errors and model order of autoregressive models fitted to the sulphuric acid in the anolyte (broken line) and the feed (solid line)

.

In this chapter the merits of the previously proposed variants have been considered via simulated case studies and real systems in the context of process engineering. In addition, a nonlinear version of SSA based on the use of kernels has been proposed. Simulation studies have shown that the nonlinear kernel-based SSA method is able to significantly reduce energy scatter compared to linear and MLP-based nonlinear version when the underlying signal is strongly harmonic. Although not explored because of limited scope of study, the results indicate that the method can be useful in other applications, for example data rectification, gross error detection and multiscale analysis.

The proposed kernel-based nonlinear SSA was subsequently extended to hypotheses testing for time series classification. Simulation studies showed that tests based on the nonlinear variant performed better than the equivalent linear formulation for certain ranges of the kernel hyperparameters. The Allen-Smith tests, where covariance matrices of the surrogates are projected onto the eigenspace of the data, performed better compared to the Elsner-Tsonis test, which provides for a direct comparison of the eigenspaces of both data and surrogates. It was found that an improved procedure for time series classification is possible using the nonlinear variations. Unfortunately, the mapping and subsequent decomposition of the covariance matrix in a high-dimensional space is implicit. Therefore, it is difficult to interpret the eigenspectra (for example, eigenpairs indicating an oscillation as in linear SSA). Further work will be directed toward extracting further useful information from these eigenspectra beyond time series identification. Finally, the potential application of linear and nonlinear SSA to real-world data was illustrated using measurements from metallurgical plants.

Chapter 5

Nonlinear Projective Methods in Process Monitoring and Diagnostics

> ...They reduce everything to a few mathematical formulas of equilibrium and superiority, of time and space, limited by a few angles and lines. If that were really all, it would hardly provide a scientific problem for a schoolboy.
>
> Carl Von Clausewitz (1780-1831)

THE continuous search for novel methods for fault detection and identification resulting from many incentives as highlighted in Chapter 1 has recently drawn attention to support vector machines as a means toward improved fault diagnosis. As explained earlier, kernel-based methods are in theory capable of better generalization, particularly as far as large systems are concerned, since their performance is not dependent on the number of variables under consideration and recent studies have underlined their promising role in diagnostic systems.

In this chapter, integration of these methods into the classical multivariate statistical process control framework is considered. In the first contribution, one-class support vector machine (SVM) classification is proposed to estimate nonparametric confidence limits for data visualization and improving graphical process monitoring charts. A residual approach to process monitoring using kernel principal component analysis is introduced that attempts to resolve the problem of interpretation and analysis of process data when a fault is detected using feature space methods. Using these limits in conjunction with biplots and standard statistics collectively constitute a powerful approach to monitoring process systems, as demonstrated by case studies on mineral processing systems. The use of nonlinear supervised feature extraction within the diagnostic framework is also investigated.

5.1 Multivariate Process Monitoring Charts Based on PCA

Process data visualization is an important tool for monitoring and diagnosis of process operations. In multivariate SPC using PCA, bivariate principal components or score plots are often used as monitoring charts to aid operators in understanding and interpreting fault information being generated on process plants (Kresta et al., 1991). These are especially useful when the underlying process dimensionality is very low (less than five), and most of the information is contained in a few latent vectors. Pairwise plots of these scores define two-dimensional "windows" on the behavior of the high dimensional process. More specifically, given a process with, say, three dominant latent variables $\mathbf{t}_1, \mathbf{t}_2$, and $\mathbf{t}_3$, bivariate score plots ($\mathbf{t}_1$–$\mathbf{t}_2, \mathbf{t}_1$–$\mathbf{t}_3, \mathbf{t}_2$–$\mathbf{t}_3$) with similar control limits similar to univariate chart for the mean of the variable or Shewhart's chart are plotted and deployed for use in monitoring. Also included on the charts are the associated reference data scores. Assuming the scores are i.i.d. samples from a Gaussian distribution, control limits on an $\mathbf{t}_i$–$\mathbf{t}_j$ plane form an ellipse, whose proper size is found using the variance information. Diagnostic capability of the score plots can be improved by including information on the squared prediction error (SPE) or Q statistic using a range chart, see Figure 2.4.

Although the multivariate normality assumption on the scores is reasonable because of the central limit theorem (Nomikos and MacGregor, 1995), finite data sizes as well as serial correlation may invalidate normality and independence assumption respectively. Instead of a hypothesized statistical distribution, an alternative is to use data-driven nonparametric approaches for density estimation (Silverman, 1986). Nonparametric methods assume no prior knowledge on the statistical nature of the data is available. Included in this group of techniques are Parzen density estimation, nearest neighbor methods, and clustering approaches (Markou and Singh, 2003).

Kernel density estimation (KDE) is a widely used nonparametric method for estimating density functions. Chen et al. (1996) proposed the use of KDE in defining the normal region associated with normal process behavior using multivariate statistical methods. Martin et al. (1996) introduced the M^2 statistic as a nonparametric-based confidence bound more suitable for complex process data than the alternative Hotelling's T^2 statistic. The statistic is obtained by integrating KDE with standard bootstrap re-sampling techniques. An important issue in KDE is the selection of the appropriate kernel bandwidth parameter, which determines smoothing properties of the obtained estimate. Chen et al. (2000) did a comparative analysis of a few of the most important methods for bandwidth selection with respect to use with complex process data in MSPC.

As remarked in Section 3.3.2, density estimation is a difficult problem particularly for small-sized data sets. In the next section a method for estimating a confidence bound using support vector classification is introduced. More specifically, the use of one-class classification methods introduced earlier are explored in the context of process monitoring. Besides the strong conceptual foundation and computational simplicity of one-class SVMs, they also appeal to MSPC monitoring charts. Since control limits are density level points, one-class SVMs are appropriate from this perspective as only the support of a distribution is computed. Whereas density estimation-based methods require computationally intensive bootstrap re-sampling to estimate the control limit, specifying the parameter ν in one-class SVMs determines the threshold level.

Although score plots are an important graphical tool for process data visualization for the purposes of plant control, their value can be enhanced by augmenting the plots with information on the process variables using the biplot methodology (Aldrich et al., 2004; Gardner et al., 2005). Not only all does such information assist with evaluating how an out-of-control sample in score plot relates to the original variables, but also how plant variables are correlated. Such graphical exploratory data analysis tools easily reveal patterns and relationships than is possible with other data analysis techniques (Everitt and Dunn, 2001). In the next paragraphs an overview of the biplot methodology is given.

The biplot was first introduced by Gabriel (1971) as a graphical tool to represent row and column effects of multivariate data in a few dimensions (typically two). Any $\mathbf{X}_{n\times p}$ matrix can be decomposed into two sets of matrices $\mathbf{G}_{n\times r}$ and $\mathbf{H}_{r\times p}$ representing row and column effects respectively:

$$\mathbf{X}_{n\times p} = \mathbf{G}_{n\times r}\mathbf{H}_{r\times p}{}' = \begin{bmatrix} \mathbf{g}_1{}' \\ \vdots \\ \mathbf{g}_n{}' \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \cdots \mathbf{h}_p \end{bmatrix} \tag{5.1}$$

where r is less than or equal to the rank of matrix $\mathbf{X}$. The biplot is a graphical representation of both the row and column effects on the same plot with an element of $\mathbf{X}$ represented by the inner product of the vectors corresponding to its row and column, that is $x_{ij} = \mathbf{g}_i{}'\mathbf{h}_j$. The inner product between two vectors can be geometrically interpreted as the product of the length of a vector and the length of the projection of the other on the first, which can be shown to approximate x_{ij}. Proportional vectors will lie in the same direction, while the zero elements of the matrix will be represented by perpendicular row and column effects.

Gower and Hand (1996) have since presented a different perspective of the concept and regard biplots as multivariate analogues of scatter plots that facilitate interpretation of multivariate relationships in the observations. It is based on the very familiar concept of canonical axes, such as the Cartesian coordinate system. Mathematically, the Gower and Hand biplot methodology is based on the traditional Cartesian representation of an p-dimensional space by p orthogonal coordinate axes. The position of a sample i relative to the axes is given by the vector sum $\sum_j x_{ij}\mathbf{e}_j$. The j^{th} Cartesian axis is the locus of $\rho\mathbf{e}_k$ for $-\infty \leq \rho \leq \infty$ and the value x_{ij} of the i^{th} sample on the j^{th} axis is given by

$$\mathbf{x}_i\mathbf{e}_j\mathbf{e}_j{}' = x_{ij}\mathbf{e}_j. \tag{5.2}$$

Therefore, the interpolation of a point $\mathbf{x}_i \in \mathbb{R}^p$ can be expressed as

$$\mathbf{z}_i = \mathbf{x}_i\mathbf{V}_r = \sum_{j=1}^{p} x_{ij}\mathbf{e}_j{}'\mathbf{V}_r. \tag{5.3}$$

The Gower and Hand methodology introduces interpolation biplot axes that allow for graphical interpolation of sample points; the j^{th} interpolation biplot axis in the space $\mathbb{R}^r$ is defined by $\rho\mathbf{e}_k{}'\mathbf{V}_r$.

Conversely, it is also possible to define biplot axes that facilitate inference of values of the original p variables in $\mathbb{R}^r$. This requires an inversion of the interpolation process and is

referred to as *prediction* in Gower and Hand's methodology. Since $\mathbb{R}^r \subset \mathbb{R}^p$, the coordinates of the projection $\mathbf{z} \in \mathbb{R}^r$ can be expressed in terms of the basis of $\mathbb{R}^p$ as

$$\mathbf{z}_i = x_i \mathbf{V}\mathbf{V}' = \sum_{j=1}^{p} x_{ij} \mathbf{e}_j \mathbf{V}_r \mathbf{V}' = \sum_{j=1}^{p} x_{ij} \mathbf{b}_j, \tag{5.4}$$

where $\mathbf{e}_j \mathbf{V}_r \mathbf{V}'$ are row vectors referred to as biplot axes that define p directions in $\mathbb{R}^r$. These biplot axes in $\mathbb{R}^r$ define a reference system similar to the Cartesian system. The position of a sample in $\mathbb{R}^r$ is given by the vector sum $\sum_{j=1}^{p} x_{ij} \mathbf{b}_j$ with non-unit vector b_k now assuming the role of $\mathbf{e}_k$. Projection onto the k^{th} biplot axes is given by

$$\frac{\mathbf{x}\mathbf{b}_k \mathbf{b}_k{}'}{\mathbf{e}_j \mathbf{V}_r \mathbf{V}' \mathbf{e}_{j'}} = \mu \mathbf{x}_k \mathbf{b}_k{}' \tag{5.5}$$

where $\mu_k^{-1} = \mathbf{e}_j \mathbf{V}_r \mathbf{V}' \mathbf{e}_{j'}$ is a normalization factor. From Equations (5.4) and (5.5) it is can be seen that, apart from a normalizing factor, projection onto the biplot axes $\mathbf{b}_k$ is similar to projection onto conventional Cartesian axes $\mathbf{e}_k$. To aid in visual interpretation, the biplot axes can be calibrated similar to Cartesian axes. However, the presence of a scaling factor in the prediction case (Equation 5.5) results in different calibrations for the interpolation and prediction cases. For a conventional Cartesian reference system $\mu_k = 1$ and, therefore, the interpolation and prediction calibrations coincide). In the following, the calibration of the biplot axes is of the prediction type, which is more interesting and relevant.

5.2 Improved Process Monitoring Charts Using SVMs

In this section one-class classification methods based on SVMs are used in defining the normal region associated with an in-control process. Three algorithms will be considered, viz. the standard one-class SVM (Equation 3.87), the generalized one-class SVM (Equation 3.98), and the ℓ_1-one-class SVM (Rätsch et al., 2002). For the generalized one-class SVM, data sampled uniformly around the in-control sample will be used as a proxy for the unknown abnormal class. In this case, the objective is to obtain a minimum volume estimation to the normal region. Data from industrial operations are used to illustrate the approach.

5.2.1 Case Study I: Platinum Group Metals (PGM)Flotation plant

Froth flotation is a well-established and important mineral processing method that is widely used to separate gangue from valuable ores. In the last couple of decades, image analysis of flotation froths has become an important element in the development of better control strategies for flotation circuits (Aldrich et al., 1997; Hyötyniemi and Ylinen, 2000; Moolman et al., 1995). Although no automated systems incorporating image analysis of froths appear to be established yet, manual control decisions are often based on visual inspection of the froth surface – a skill which largely depends on operator experience and therefore potentially unreliable. To assess the applicability of the one-class SVM nonparametric confidence limit methodology in industrial practice, data from two industrial flotation plants previously investigated by Moolman et al. (1996) were analyzed. Images of the froth surface sampled at regular intervals were used to extract a set of statistical features based on the gray-level

patterns in the digitized images. In this first example, feature extraction from textural information of froth structures generated on a platinum flotation plant in South Africa is considered. The data are composed of five image features characterizing the froth (Aldrich et al., 2004):

1. SNE (i.e. small number emphasis, inversely related to bubble size),
2. ENTROPY (high values representing more complex images than low values),
3. INERTIA (a measure of the number of local variations in the image),
4. LOCHOM (local homogeneity), and
5. GLLD (gray level linear dependencies).

A plot of the scores obtained by projecting the observed data onto the two leading principal directions explaining 95% of the variation is shown in Figure 5.1.

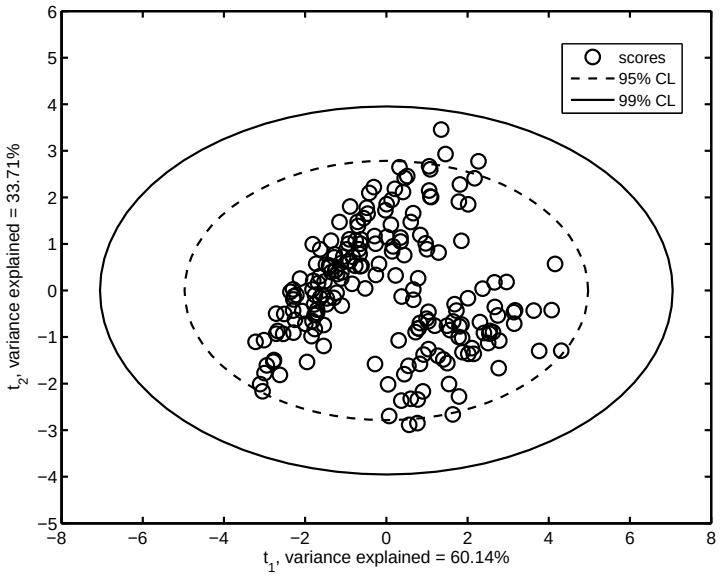

Figure 5.1: A 2-dimensional PCA projections of textual taken from digitized froth images from a PGM flotation plant. Superimposed are the control limits assuming multivariate normality

The graphical summary shows that the latent variables in the low-dimensional space can be used instead of the five original variables, since the residual variance captured in the other directions is small ($\approx 6\%$) and probably a result of high frequency components in the data.

Multivariate statistical monitoring requires a confidence region to be defined on the scores plane. In this case, an elliptical region can be derived (assuming normality and appropriate scaling) and superimposed on the plane of the scores plot as shown in Figure 5.1. For the PGM data the scores samples are not evenly distributed. Hence, although an elliptical confidence region is still feasible, the hypothesized underlying distribution is inconsistent with the data, resulting in large empty regions of the scores plane considered as part of the normal region.

Alternatively, Tukey's bagplot or α-bags could be considered since it is a useful tool for visualizing the central tendency, dispersion, correlation, skewness and tails of bivariate data in a similar way as a boxplot does for univariate data (Rousseeuw and Van Driessen, 1999). An α-bag is a bivariate analogue of the univariate boxplot and can be defined as a contour enclosing the exact innermost α% of samples in a bivariate scatter plot. Figure 5.2 shows an 80% bagplot superimposed on the PCA scores plane.

Also shown superimposed is predictive biplot information, which clearly identifies the relationships between the samples and variables. A unit aspect ratio is enforced on the plot to avoid distortion of information when projecting a sample onto the biplot axes. Note that the usual Cartesian axes are not calibrated or labeled. As argued in Gower and Hand (1996), these axes are only useful scaffolding axes but have little value in providing information that is of relevance. Instead, the non-orthogonal, labeled and calibrated biplot axes are used to provide information on the samples. Also, the relationships between the variables can be approximated in a similar way as in the Gabriel biplot. Thus, it can be seen that the SNE and INERTIA are highly and negatively correlated, that is an increase in SNE results in a decrease of INERTIA and vice versa. However, a change in either of these has little effect on ENTROPY or LOCHOM, as discussed in more detail in Aldrich et al. (2004). Both SNE and INERTIA are related to local variations in the image and are therefore related to the bubble sizes in the images and these two features are also seen to be the best discriminants between the clusters in the data. Physically, the two clusters in the data are related to different operating conditions. The smaller cluster represents operating conditions where the froths consisted of ellipsoidal bubbles, heavily loaded with gangue minerals that gave the images a lighter appearance than those of the larger cluster, where the froths had more spherical bubbles and contained less gangue minerals.

Unfortunately, the limitations imposed by uneven data distribution are still not completely solved, with the bagplot enclosing regions that clearly are not populated by the data. Hence, out-of-control observations may still remain undetected on the basis of the graphical information.

Instead of bagplots, it is proposed to define a confidence region on the scores plane using one-class SVMs. As in bagplots or standard MSPC, a user-defined parameter (ν) determines the size or limits of the normal region of plant behavior. More specifically, choice of the parameter defines a quantile of the underlying distribution where most of the data lies. Therefore, the "correct" value to use depends on the allowed quality tolerances for the system under consideration. In particular, if the penalties of incorrect quality specifications of the final product are very high, one would expect tighter confidence bounds.

In Chapter 3, a number of one-class SVM algorithms were highlighted. Three of these algorithms were used to estimate the quantile region containing 80% of the data (in feature space, after mapping) specified by the choice of $\nu = 0.2$. Figures 5.3, 5.4 and 5.5 show respective results obtained for the standard one-class SVM of (Schölkopf et al., 2001), the generalized one-class SVM (Schölkopf et al., 2000b), and the ℓ_1-norm one-class SVM (Rätsch et al., 2002). As the figures indicate, no substantial differences are observable among the decision boundaries obtained by each method for this data. Also, while a closed boundary for the normal class was obtained, sparsely populated regions in the input space within the bounded region were assigned a negative weight. That is, if a new sample

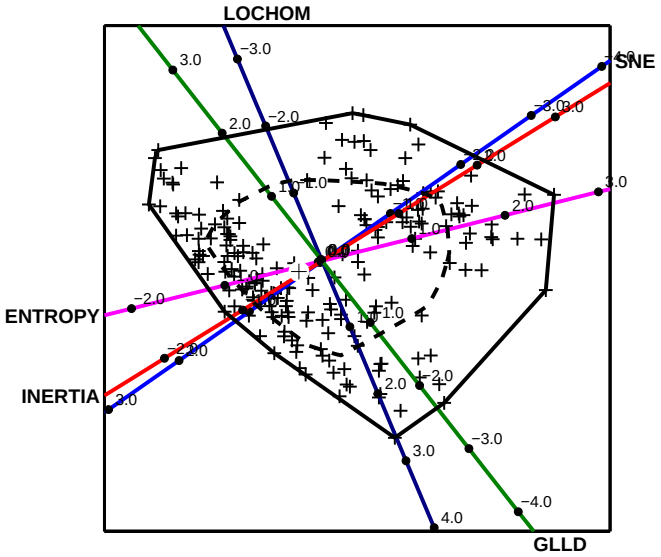

Figure 5.2: Estimating a confidence region of the 2D principal components using bagplots or bivariate bags. In this case a bagplot containing approximately 80% of the data ($\alpha = 80\%$) is shown (dashed line). The convex hull (solid line) is a fence separating possible outliers in the data.

pattern falls into such regions an alarm is triggered. While such regions may probably be a stochastic effect of data generation, for normal regions defined using a large historical database, they provide useful information on the preferred operating regime associated with the plant components or dynamics. As such, when the graphical charts indicate an alarm for a pattern falling within the enclosed or inner outlier regions, it may be valuable to trace the offending data and assess whether any significant differences exist with respect to quality when compared to other normal patterns. For other incoming data falling outside the defined region a change in process conditions or equipment parameters is indicated. Back projecting the scores for the faulty data onto the biplot axes potentially gives timeous insight into the variable(s) responsible for the shift.

5.2.2 Case Study II: Monitoring of a Calcium Carbide Furnace

Aldrich and Reuter (1999) have considered the operation of a calcium carbide furnace that could be represented by daily averaged measurements of ten process variables viz. furnace load, electrical power consumption, electrode resistance, lime additive, charcoal, coke, anthracite, and three variables characterizing lime quality. The performance of the furnace was characterized by a quality index that represented the product of the production rate and the grade of the calcium carbide.

A high overall furnace load, combined with high loads of lime, charcoal and coke produced high quantities of high grade product and hence represented the normal operating region of the furnace. These four variables were highly correlated in the available process data and fault conditions were characterized by lower values (loads) on all of them. In contrast, the power consumption, electrode resistance and anthracite were weakly correlated with

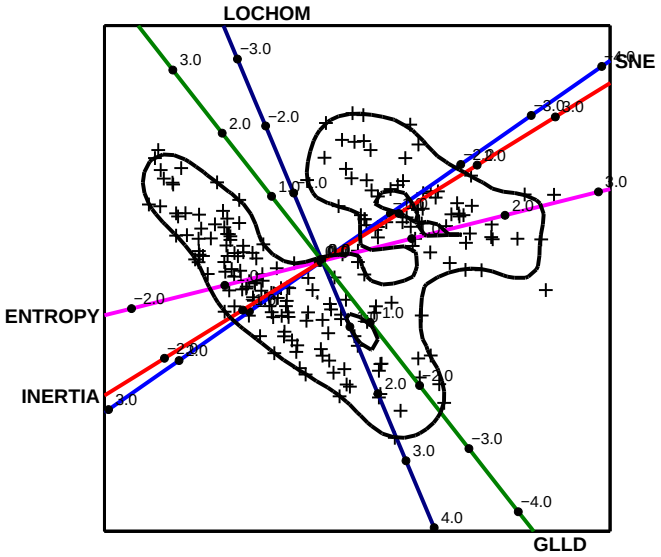

Figure 5.3: Multivariate score plot with PCA biplot axes and an 80% quantile estimate based on standard one-class SVM. The hyperparameters values used were $\nu = 0.2\%$, and $\sigma_{\text{RBF}} = 0.8$.

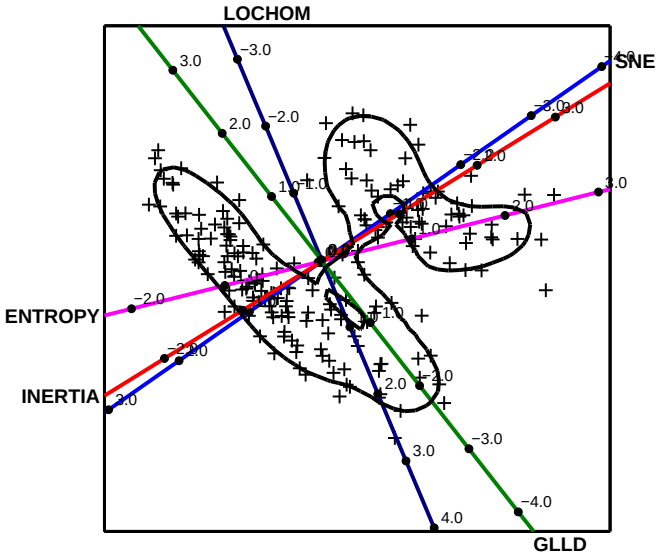

Figure 5.4: Multivariate score plot with PCA biplot axes and an 80% quantile estimate based on generalized one-class SVM. The hyperparameters values used were $\nu = 0.2\%$, and $\sigma_{textRBF} = 0.8$. The outlier class was generated by uniformaly sampling 500 points from a hypercube containing the scores.

the product grade and quality, while the lime quality appeared to have a negligible influence on the performance of the furnace.

Figures 5.6 and 5.7 are plots of the scores of the process variables obtained from the data with classical and one-class SVM-based confidence limits superimposed respectively.

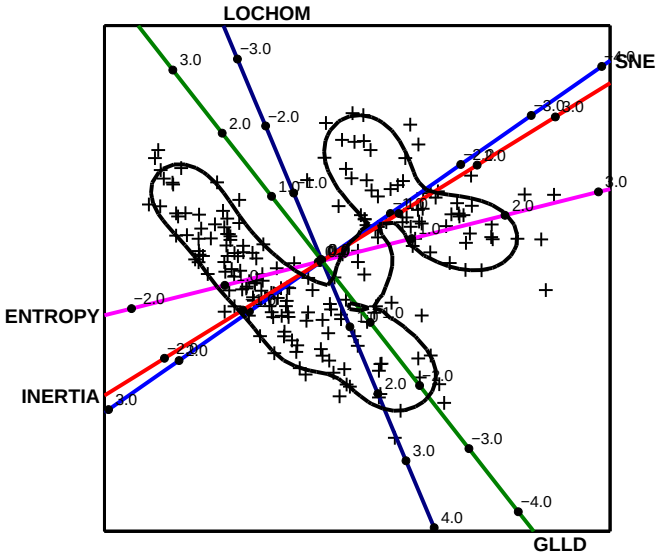

Figure 5.5: Multivariate score plot with PCA biplot axes and an 80% quantile estimate based on ℓ_1-norm one-class SVM. As in the preceding cases, hyperparameters values used were $\nu = 0.2\%$, and $\sigma_{\text{RBF}} = 0.8$.

In these figures, desirable and faulty operating conditions are indicated by '+' and 'o' markers respectively superimposed on the principal component scores. When new data known to represent faulty operating conditions are projected onto these reference models, the confidence bounds of the linear PCA score plot yield a higher fraction of false positives (38%) compared to the one-class SVM approach (<20%).

5.3 Fault Detection Using Kernel PCA: A Residual Analysis Approach

In kernel PCA (Section 3.3.1), the mapped data belongs to a high-dimensional space $\mathcal{H}$ (potentially infinite for some kernel functions such as the Gaussian kernel). Therefore, the dimensionality of useful projections can be much higher than the dimension of the input space (Burges, 2005). This has important implications in the case of using kernel PCA for process control. In particular, below it is proposed to use kernel PCA for feature extraction only. This is in contrast to previous studies that were more similar to the classical MSPC approach, where the focus was on the derivation of monitoring statistics (T^2 and squared prediction error) from the embedded data in feature space(Choi et al., 2005).

MSPC efforts are then focused on the input space residuals. This requires a mapping of the projected feature subspace back to input space – the so-called preimage problem (Schölkopf et al., 1999). Mapping feature space data back to the input space is an ill-posed problem, because some of the points in the feature space have no corresponding exact preimage in the input space. Approximate preimages can be found using different proposed algorithms such as fixed-point iteration (Schölkopf and Smola, 2002), learning a

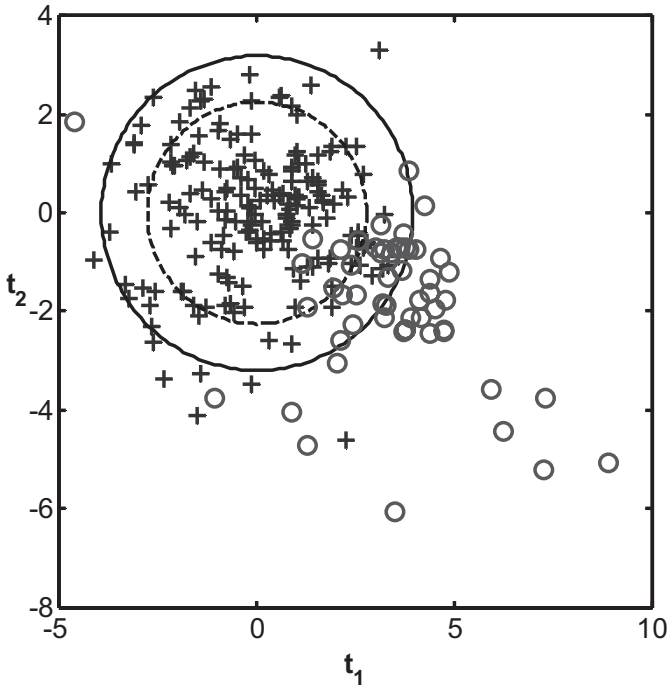

Figure 5.6: PCA scores plot for data taken from a DCM furnace. Superimposed are 95% (dashed) and 99% (solid) confidence limits determined using the normal ('+') data. The fault conditions are indicated with circle ('∘') markers, a relatively high proportion of which fall within the 99% limit and, therefore, will not trigger an alarm.

mapping function using any regression algorithm (Bakir et al., 2004; Bishop, 1995), and multi-dimensional scaling approaches (Kwok and Tsang, 2004).

With the proposed strategy, nonlinear features ($\mathbf{F}$) are first extracted from the data matrix ($\mathbf{X}$) representing normal operating conditions. Feature extraction can be unsupervised (kernel PCA) or supervised (kernel FDA). The data are subsequently reconstructed by mapping the features ($\mathbf{F}$) back to ($\mathbf{X}$). This approximation gives the reconstructed data ($\tilde{\mathbf{X}}$). In the case studies considered below, the mapping was done with the multi-dimensional scaling approach (Kwok and Tsang, 2004) although any other suitable model could have been used.

The residuals ($\mathbf{e}$) arising from the difference between ($\mathbf{X}$) and $\tilde{\mathbf{X}}$) are then considered further. Linear principal components are extracted from the residual matrix, giving a score matrix ($\mathbf{T}$), as well as other statistics, such as Hotelling's T^2 or squared prediction errors (Q statistics). The general strategy for fault detection and identification with kernel methods is summarized in Figure 5.8.

In addition, nonparametric confidence limits are generated by means of one-class SVM methods described above. The identification of faults is facilitated by means of Gower and Hand biplots. The strategy is general and different variants can be used by substituting the elements of the strategy with different ones. For example, instead of biplots, contribution plots can be used, or instead of generating confidence limits with one-class SVM methods,

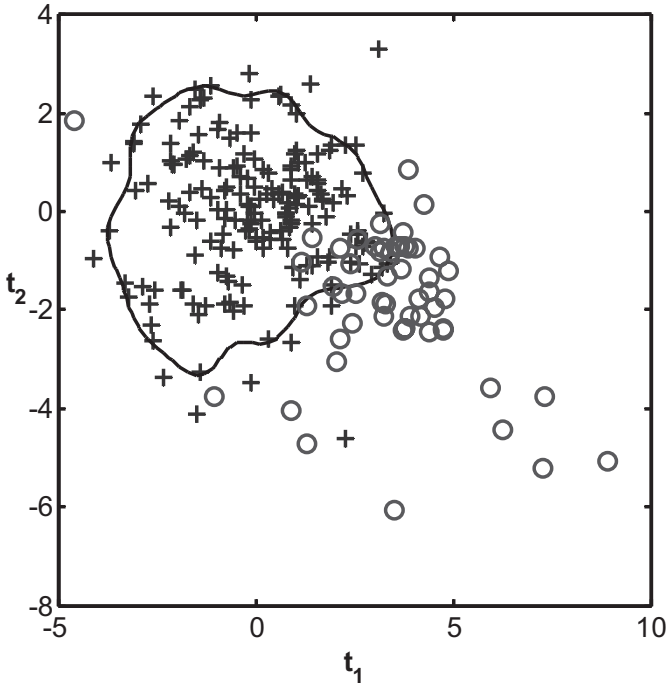

Figure 5.7: Similar plot to Figure 5.6 but instead of PCA-based control limits, an estimated support of the distribution for $\nu = 0.2$ is shown. The detection of the fault condition is improved as the fraction of faulty data lying outside the estimated 80% is greater than in the preceding figure. However, the false alarm is also slightly increased.

α-bags or other methods can be used. The same goes for the model used to reconstruct the data, the method used to extract the features from $\mathbf{X}$ in the first place. This strategy is considered by means of case studies as follows.

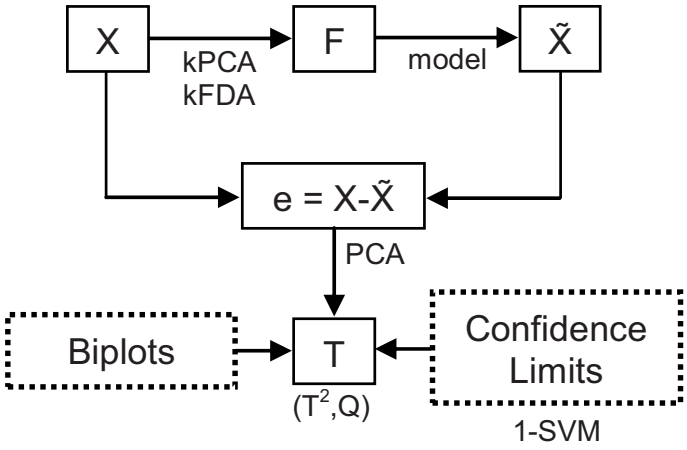

Figure 5.8: Schematic representation for residual-based fault diagnosis using kernel learning methods

In the case studies below, various aspects of the strategy outlined above are considered. In

the first case study on simulated data, the merits of feature extraction with kernel based methods are compared with other methods (principal components and curves). In the second case study, data from the calcium carbide furnace used above is considered.

5.3.1 Case Study I: Simulated System

To illustrate feature extraction with the proposed kernel-based method, a simulated system used by Dong and McAvoy (1992) is considered. The system is driven by a single variable (t) which is inaccessible and the only information available are three measurements satisfying

$$\begin{aligned} x_1 &= t + \varepsilon \\ x_2 &= t^2 - 3t + \varepsilon \\ x_3 &= -t^2 + 3t^2 + \varepsilon_3, \end{aligned} \tag{5.6}$$

where $t \in [0.1, 1]$ is sampled from a uniform distribution and $\varepsilon \sim \mathcal{N}(0, 0.02)$. A fault condition is introduced by inducing small changes to the measured variable x_3 and the abnormal situation is then defined as

$$\begin{aligned} x_1 &= t + \varepsilon \\ x_2 &= t^2 - 3t + \varepsilon \\ x_3 &= -1.1t^2 + 3.2t^2 + \varepsilon. \end{aligned} \tag{5.7}$$

The reference data for deriving multivariate statistical process monitoring charts consisted of 100 samples collected before the occurrence of any fault condition. A further 100 samples were collected in the presence of the fault condition. The two sets of data are shown in Figure 5.9. Although the data sets are different, it is difficult to detect the fault condition visually by monitoring the evolution of the process using a 3D scatter plot. MSPC approaches based on classical principal component analysis are also inadequate, since the correlations between the variables are nonlinear, as shown in Figure 5.10.

Dong and McAvoy (1992) proposed a method that makes use of principal curves and auto-associative multilayer perceptrons for nonlinear process monitoring. The principal curve method of Hastie and Stuetzle (1989) was used to find nonlinear scores, while a multilayer perceptron network was used to find both forward and reverse nonparametric mappings between the scores and original data.(A nonlinear principal loading is not defined explicitly for the principal curve method.) Using the normal data a principal curve was found and Figure 5.11 is a plot of the squared prediction error or Q statistic derived for their method. It can be seen that the onset of the fault condition is detected.

Results of the proposed kernel PCA based approach on the same data are shown in Figure 5.12, where kernel PCA was used to extract nonlinear features from the data. Both the T^2 and SPE statistics indicate a process shift in the data sampled in the presence of the fault condition. This is further emphasized in the scores plot in Figure 5.13, where the ellipses indicate the confidence limits for the principal scores. The deviation in the fault condition data is shown by the points lying outside the limits. Figure 5.14(a) shows an illustration of the leading principal direction (dashed line), the one-dimensional principal curve (solid line),

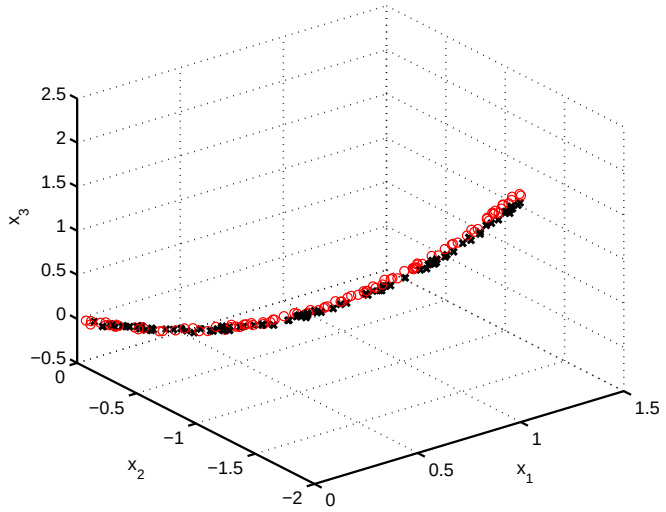

Figure 5.9: A 3D scatter plot of data sampled from the Dong–McAvoy simulated system under normal ('+') and abnormal ('∘') operating conditions, i.e. Equations 5.6 and 5.7 respectively.

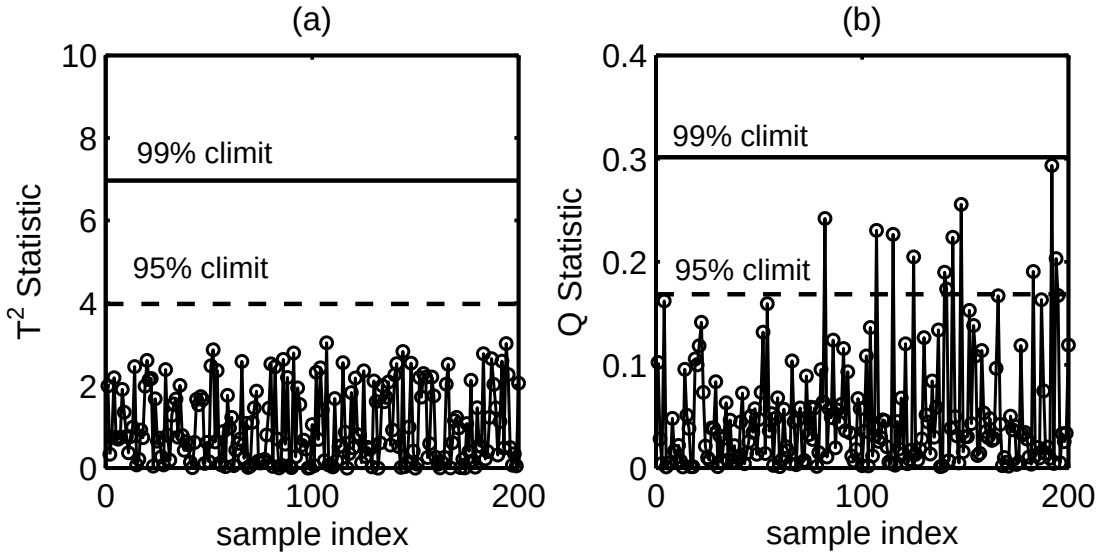

Figure 5.10: Hoteling's T^2 and (b) Squared prediction error or Q statistics for both the normal data (first 100 samples) and the fault condition data (last 100 samples) calculated for the Dong-McAvoy simulated system. The superimposed horizontal lines indicate the 95% (dashed line) and 99% (solid line) confidence limits for each statistic. In both cases, the fault condition remains undetected at the 99% confidence level.

and in Figure 5.14(b) the kernel-based principal component (solid line). The kernel-based principal component and the principal curve are virtually identical.

The advantage of the proposed method compared to the principal curves approach is that nonlinear optimization is avoided in kernel PCA and, hence, potentially suboptimal solutions. Also, principal curves require prior specification of the number of features to be

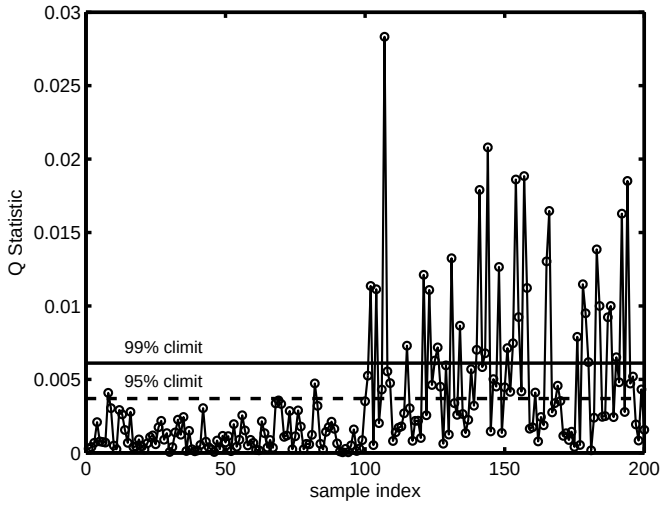

Figure 5.11: SPE monitoring chart using principal curves-multilayer perceptron method. The onset of the fault condition after the 100^{th} is clearly observed.

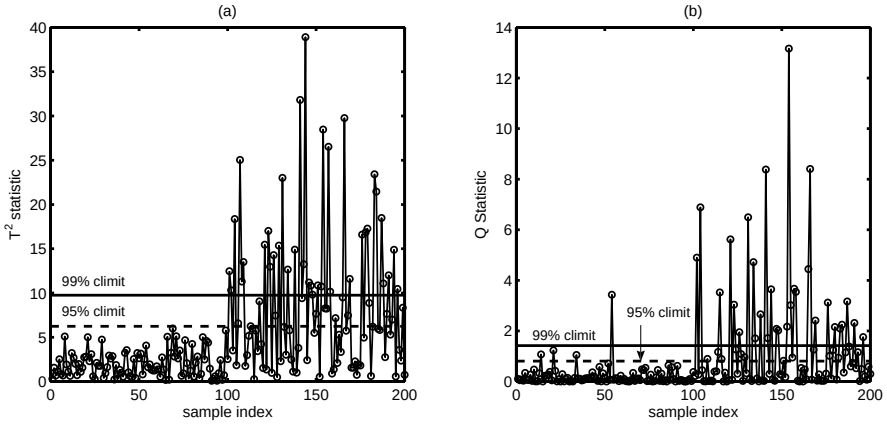

Figure 5.12: (a) Hoteling's T^2 and (b) SPE statistics for the simulated system after extracting dominant nonlinear features using kernel PCA, and performing linear PCA on the residuals. Similar to Figure 5.11, the onset of the fault condition can be distinguished using either of the statistics.

extracted. The main disadvantage of the kernel PCA method is the lack of clear geometric interpretation of the features extracted in the input space. In the experiments conducted, the approximate pre-images were sensitive to the reconstruction method used.

5.3.2 Case Study II: Monitoring of a Calcium Carbide Furnace

The calcium carbide furnace data considered earlier in Section 5.2 is re-visited. Kernel PCA can be used to remove coherent structures in the data and linear PCA analysis on the residuals can be expected to yield reliable confidence limits on the scores based on the classical Gaussian assumption. However, it may still be necessary to use advanced quantile

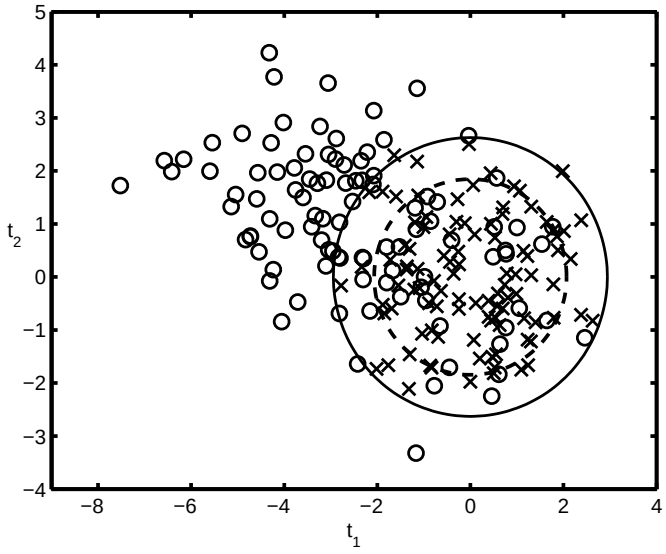

Figure 5.13: Scores plot of residuals from the simulated data in Fig. 2 after kernel PCA-based feature extraction. Superimposed on the plot are the 95% (dashed line) and 99% (solid line) confidence limits. The normal data are indicated by '+' and the fault condition data by 'o'.

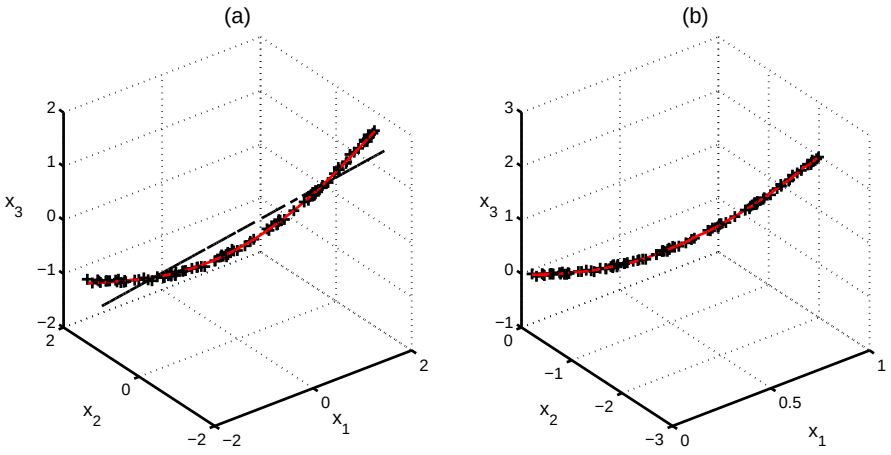

Figure 5.14: (a) Nonlinear PCA using principal curves. The solid line is the 1st principal curve and the dashed line is the first linear principal component analysis. (b) Nonlinear principal component obtained on performing kernel PCA with a Gaussian kernel of unit width. The four leading features were retained in the feature space and input space reconstruction was done using a multidimensional scaling approach (Kwok and Tsang, 2004).

estimation methods, especially when the penalty of incorrectly classifying a negative sample as positive is higher than the other way round. The decision on which confidence estimate to use is then dictated by strategic and operational considerations. Figures 5.15 and 5.16 show the 80% bagplot and one-class SVM quantile support respectively on the residuals of normal operating data, after removal of any nonlinear structure with kernel PCA.

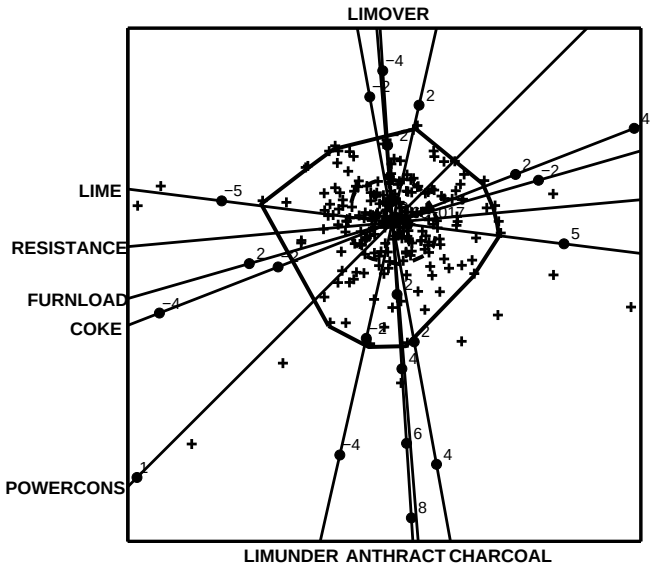

Figure 5.15: PCA biplot with an 80% α-bag region

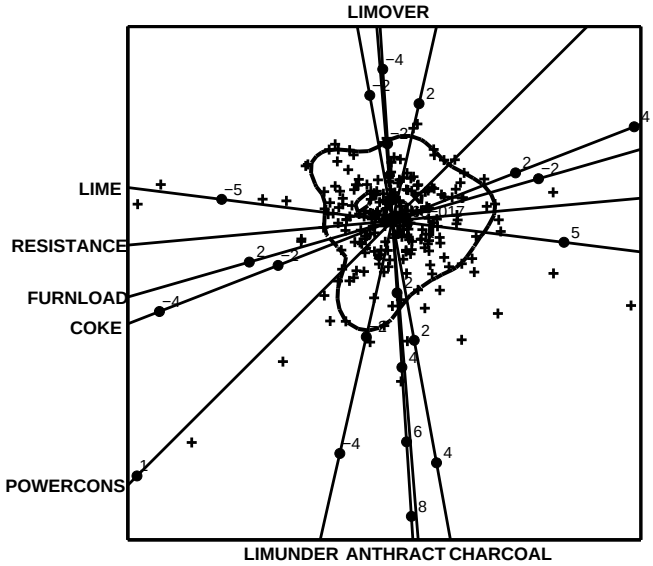

Figure 5.16: PCA biplot with an 80% quantile estimate region

5.4 Process Monitoring By Use of Discriminant Analysis

The previous sections considered feature extraction using unsupervised learning within the context of process monitoring, fault detection and identification. In this section, feature extraction using supervised learning is investigated for similar purposes.

5.4.1 Case Study I: Platinum Group Metals Flotation Plant

Inclusion of class information in nonparametric modeling is known to improve descriptive and discriminative characteristics of data (Mika et al., 2003). To see the effect of class information on the interpretation and diagnostics, the data from the platinum group metals (PGM) flotation plant described in Section 5.2.1 were grouped into three classes on the basis of the visual appearance of the froth and associated reagent additions. Figure 5.17 shows the principal component scores of the pre-processed froth features, where the class information (froth types) is superimposed on the score plot using different markers. In this case, Class 2 represents desirable operating conditions, while Classes 1 and 3 represent fault conditions. Class 1 is similar to Class 2, except that the froth is too viscous for optimal recovery of the valuable concentrates. Although Class 3 is relatively easy to detect (representing depleted froths that differ markedly from froths in the other two classes), Classes 1 and 2 overlap, which makes detection of the fault condition represented by Class 1 difficult.

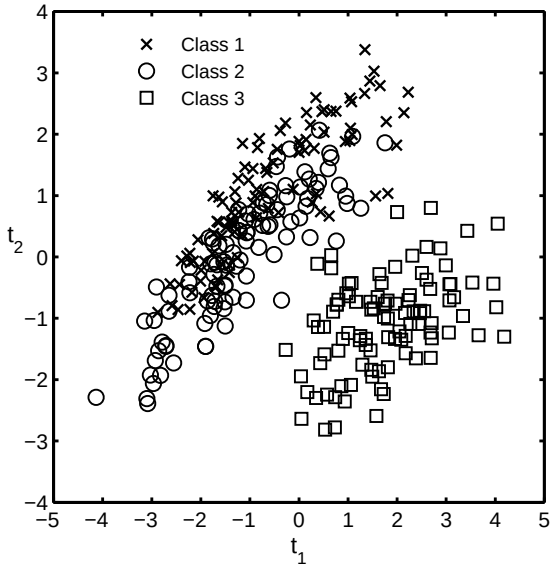

Figure 5.17: PGM flotation data principal components with class information superimposed

Figure 5.18 shows the separability of the different classes based on both linear and nonlinear discriminant analysis. In the latter case Gaussian kernels were used, where the optimal parameter for the kernel width was determined using a grid search as follows. The data were split into training and testing sets. A kernel FDA model was found for each kernel width parameter over a predefined grid, and the effectiveness of the model was validated using a support vector multiclass classifier with a linear kernel inverse model over the test set. The inverse model was used to reconstruct the original features of the data and gives an indication of the information preserved by the discriminant models. The results of the grid optimization procedure are shown in the bar plot of Figure 5.19. The numbers in the radial basis function kernels indicate the widths of these kernels, for example rbf4 is a kernel with a width of 4. Although Class 3 information was also considered in the development of the kernel-based model, this is not necessary if a linear method can be used to separate

Classes 2 and 3. Apart from the computational cost saving, it was also observed that leaving out data that can be distinguished using linear methods improved the performance of the support vector machine.

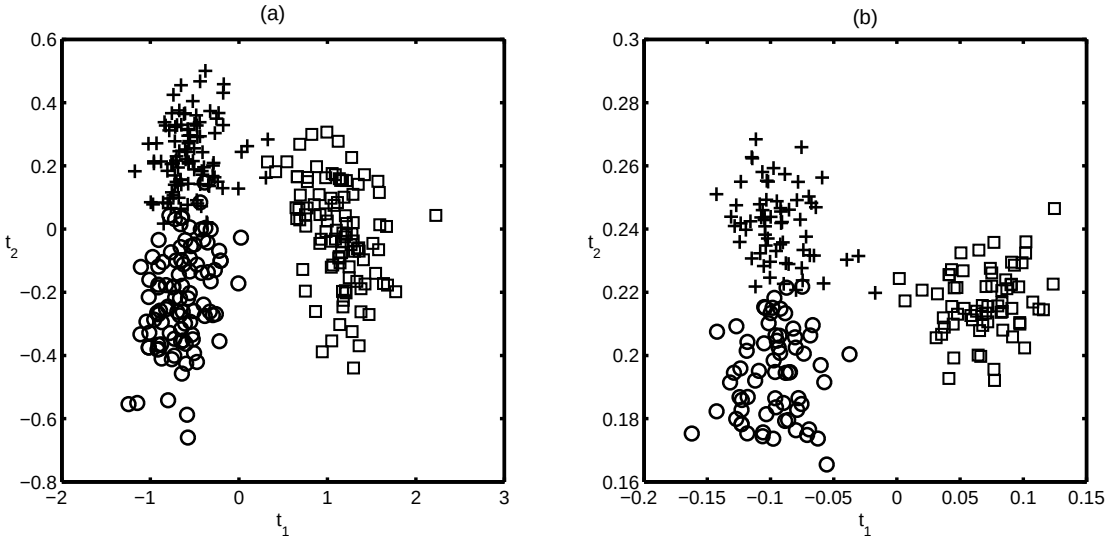

Figure 5.18: (a)Linear and (b) nonlinear discriminant analysis of PGM data. A Gaussian kernel of width=8 determined by model selection via a grid search (see Figure 5.19)

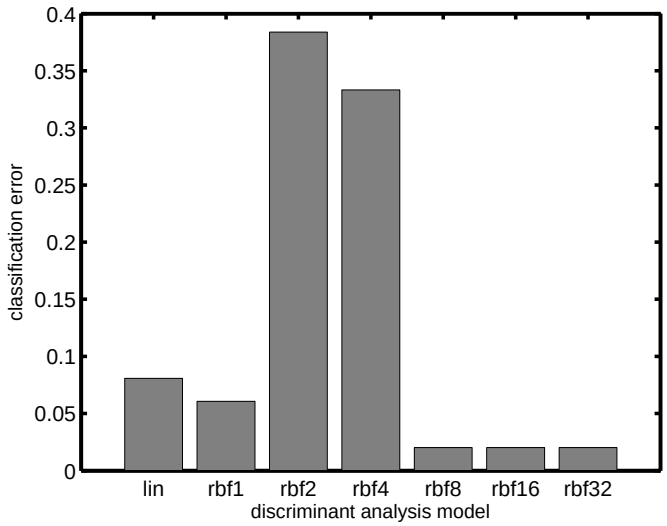

Figure 5.19: Model selection via a grid search an using independent test set for nonlinear discriminant analysis on the PGM data.

5.4.2 Case Study II: Copper Flotation Plant

These data also represent digitized froth flotation images, but this time collected from a copper flotation plant. The data were grouped into five different classes on the basis of 10

features extracted from textural information (Moolman et al., 1995). A similar approach as in the previous example above was used to determine the optimal kernel parameter. Also, one of the classes (Class 5) was not considered in the reported results, because it could be easily separated from the rest of the data on the basis of linear correlations. A typical image of each class is shown in Figure 5.20. Class 1 represents an ideal froth structure with bubbles well-loaded with minerals. Class 2 represents a deep, well-drained (dry) froth with a polyhedral froth structure, while Class 3 represents a tough froth with an ellipsoidal structure, possibly caused by too low a pulp level, too high a specific gravity or flotation of a particular type or size of particles. Class 4 represents an excessively stable, stiff froth possibly attributable to low pulp levels and/or low frother levels. In Figures 5.21(a–b) are the feature maps derived with the linear and nonlinear methods. Two of the classes (Classes 3 and 4) could not be separated by either approach, while the kernel-based method was better able to separate Classes 1 and 2 than the linear method. The optimal kernel FDA model was determined via cross-validation over a grid of hyperparameter values, the results of which are shown in Figure 5.22.

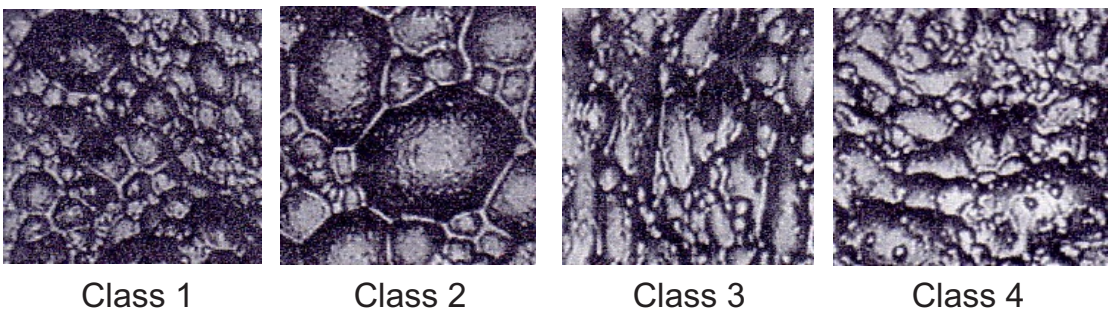

Figure 5.20: Classification of copper froth images.

5.4.3 Case Study III: Monitoring of a Calcium Carbide Furnace

To investigate the effectiveness of supervised feature training on the data from the calcium carbide furnace, it is necessary to group the data according to a quality variable. For discriminant analysis, a discrete version of the quality index was considered, with the index indicating 'low' (Class 1), 'medium' (Class 2) or 'high' (Class 3). Class 3 represented desirable operating conditions, that is where both the production rate and product grades were high, whereas Classes 1 and 2 could be treated as progressively severe fault conditions. Linear and kernel-based discriminant models were constructed from the data, using a similar approach as in the previous examples. The respective feature maps are shown in Figure 5.23 and 5.24. Visual inspection indicates that somewhat better separation between the three classes could be possible with the nonlinear feature map.

5.4.4 Case Study IV: Monitoring of an Industrial Liquid-Liquid Extraction Column

Liquid-liquid extraction systems are important mass transfer operations in the process industries, particularly where azeotropic, temperature-sensitive, and other refractory systems

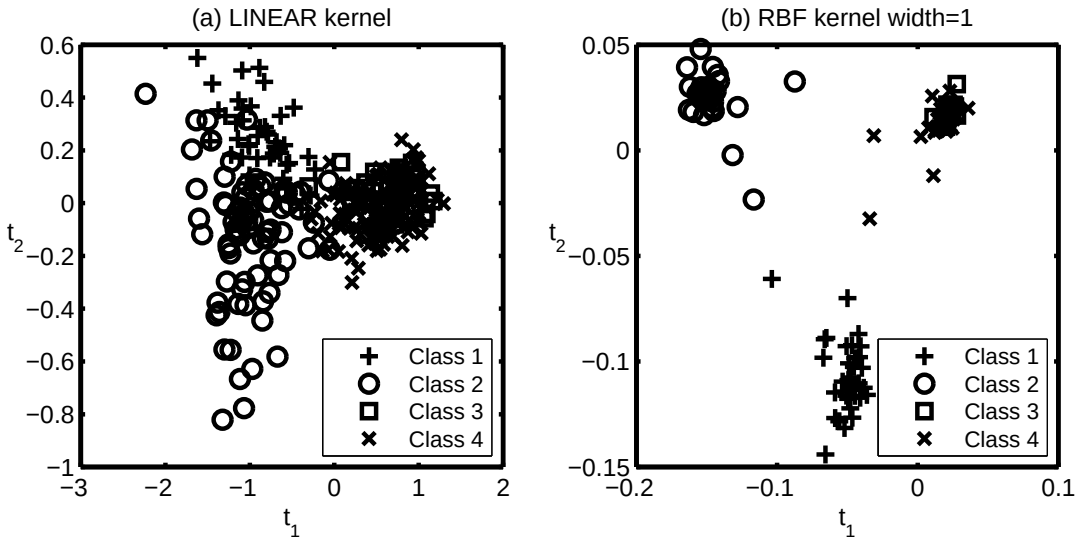

Figure 5.21: Scatter plots of the copper froth image data onto the leading (a) linear and (b) nonlinear supervised features. A Gaussian kernel of unit width determined via fivefold cross-validation on a grid of values as indicated in Figure 5.22

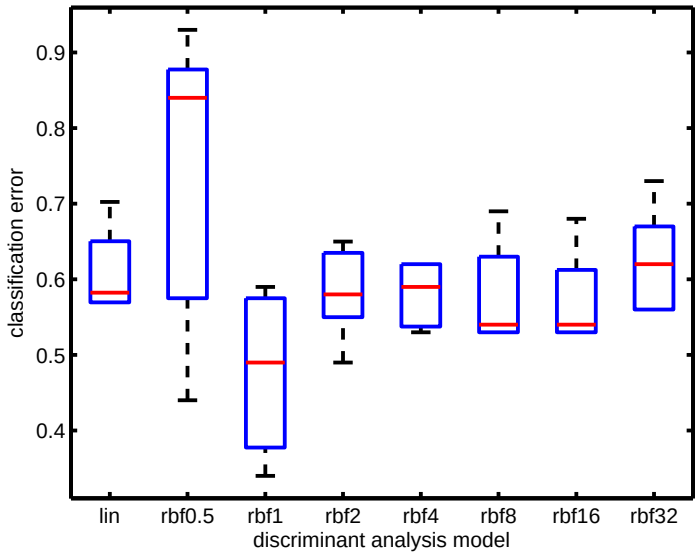

Figure 5.22: Boxplot of the performance of different discriminant analysis models over 5-fold cross-validation sets. The model using a Gaussian kernel of unit width gave the best classification error of 0.34

are concerned. The successful operation of liquid-liquid extraction columns is intricately related to the hydrodynamic and mass transfer regimes induced by column design and configuration. At present, the influence of column geometry and rheological characteristics of multiphase extraction systems are not well understood, which leads to systems whose dynamic behavior is difficult to model from first principles. This lack of fundamental

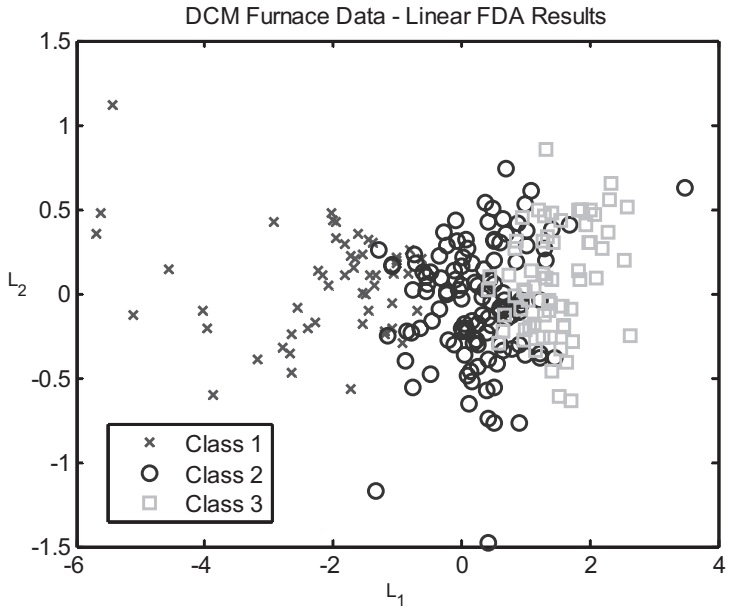

Figure 5.23: Features extracted with linear discriminant analysis for DCM furnace data

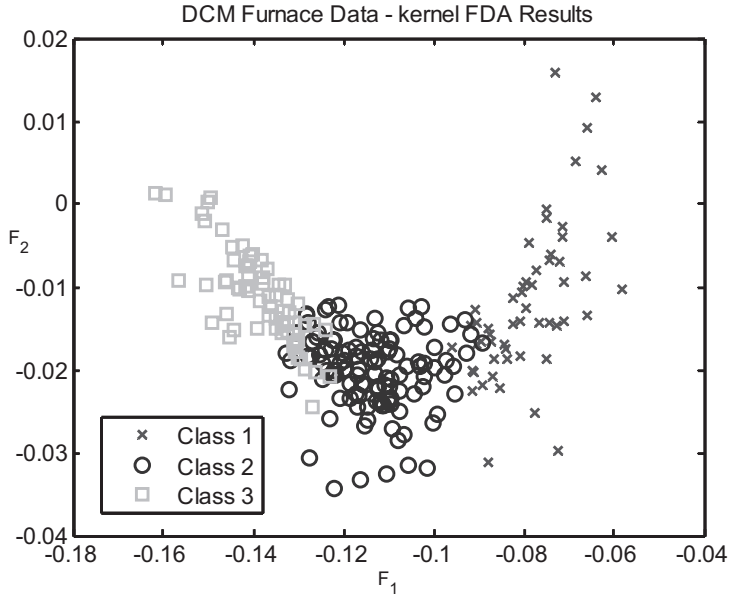

Figure 5.24: Features extracted with kernel-based nonlinear discriminant analysis

knowledge often results in inexact and over-designed columns with possibly less than desirable performance characteristics. These limitations are aggravated in environments where a premium is placed on more flexible operation, such as in the fields of fine chemistry, pharmaceutical production and the remediation of wastewater.

Various studies, initiated in response to the above mentioned problems, have indicated that it is possible to capture the dynamical properties of extraction processes with nonlinear

models (Aldrich and Slater, 1995, 2001; Boger and Ben-Haim, 1992; Giles et al., 1996; Woinaroschy, 1998). Although neural networks are by no means the only class of models that could be used in this context, they have been popular historically, in part owing to their widespread support by analytical and process control software, as well as their ease of use as rapid prototyping tools. For example, Chouai et al. (2000) have shown that neural networks could be used to capture the complex, time-variant dynamics of pilot-plant scale pulsed liquid-liquid extraction columns, fundamental modelling of which would otherwise require excessive computational effort that might defeat attempts to track continuously varying process conditions.

Most of these models are constructed as dynamic predictive models that are essential building blocks in advanced model-based control systems. Unfortunately, relatively little attention has been paid to alternative models that could be used in process monitoring, fault detection and fault identification on solvent extraction plants. Owing to their complexity, these models may not be accommodated as readily by the existing linear versions of multivariate methods, such as principal component analysis and partial least squares.

Below, nonlinear (kernel) discriminant analysis is applied in monitoring of a large industrial liquid-liquid extraction column previously studied by Aldrich and Slater (2001) by use of the following approach:

Process monitoring using discriminant analysis

1. Train a kernel Fisher discriminant object using historical data.
2. Calculate scores by projecting the data onto the kernel Fisher discriminant basis.
3. Establish normal operating conditions (NOC) for the preferred class.
4. Learn a (nonlinear) map from scores to original input variables for the NOC data.
5. Derive contribution plots for the NOC data by calculating the discrepancy between the expected values and those obtained from the regression model. The averaged errors for each variable constitute the basis for comparison with future data points for the purposes of fault diagnosis.

Although the inverse mapping in step 4 can be performed by any suitable model, a kernel ridge regression model (Cristianini and Shawe-Taylor, 2000) is used in the demonstration that follows.

Plant data description

A total of 1329 daily values of stream flow rates (x_1, x_2, x_3, x_4), degree of impurities (x_5) and temperatures (x_6, x_7, x_8, x_9) were collected over a period of five years together with the viscosity index of the product (VI), Figure 5.25. The temperature gradients and flow rates in the column were controlled in order to maintain a product of constant quality and composition, despite process disturbances associated with changes in the compositions of the feed streams.

The viscosity index (VI)is an important process quality indicator that relates the effect of temperature variations and other changes in process conditions to the viscosity of the oil, as indicated in Figure 5.25.

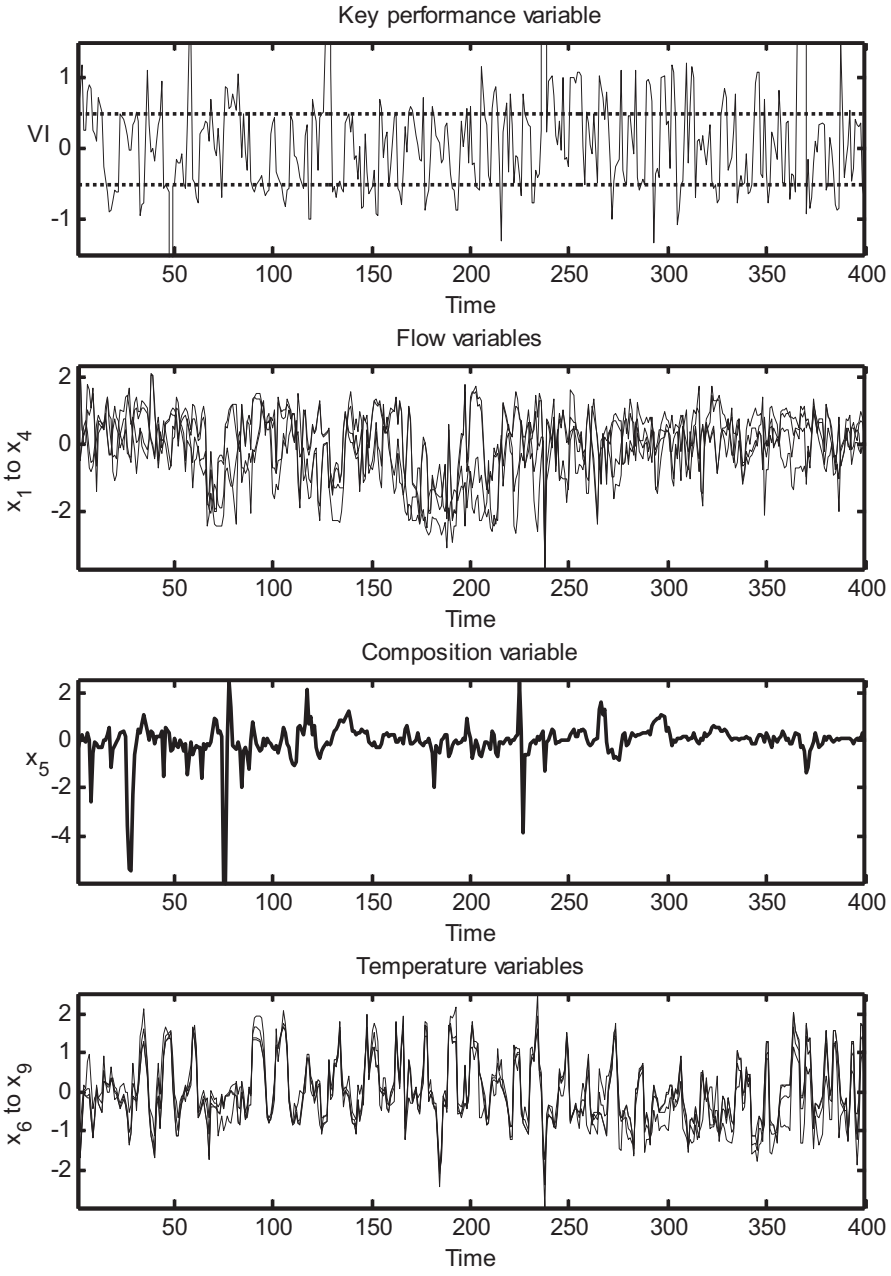

Figure 5.25: Sample variable measurements of the liquid-liquid extraction system: (a) controlled variable, (b) manipulated flow variables, (c) manipulated composition variables, and (d) manipulated column temperature variables.

Normal operating conditions (NOC) can be defined as those conditions where the quality indicator (VI) remains within specified upper and lower limits, indicated by the horizontal

broken lines. On this basis, the observations of the process variables $\mathbf{x} = (x_1, x_2, \ldots, x_9)$ could be grouped into three different categories: $\{C_i = \text{LOW, NORMAL, HIGH}\}$, where LOW indicated process conditions associated with abnormally low viscosity indices, NORMAL indicated normal operating conditions, and HIGH indicated conditions associated with abnormally high viscosity indices. Plant operation is controlled by ensuring that the viscosity index VI is maintained within the normal process limits. This is accomplished by monitoring and manipulating the process variables and disturbances related to the quality variable. When the quality or key performance variable (VI) is affected by more than a few variables, as is often the case, manual control becomes difficult without the aid of a process model.

Results and discussion

Figure 5.26 is a plot of the data $\mathbf{x}$ in feature space, after extracting the feature variables, L_1 and L_2 from the data $(\mathbf{x}_i, C_i)$ via linear discriminant analysis (equivalent to a Bayes optimal classifier), according to Equation (2.13) in Section 2.3.3. The two features L_1 and L_2 (linear combinations of the original variables $\mathbf{x}$ that maximize the separation between the three clusters shown in Figure 5.26) do not allow sharp discrimination between normal and abnormal process conditions, as the overlap between the three groups is significant. Quantitatively, the three groups could be classified with an overall accuracy of 66%, as indicated in Table 5.1 (Fisher's linear discriminant analysis).

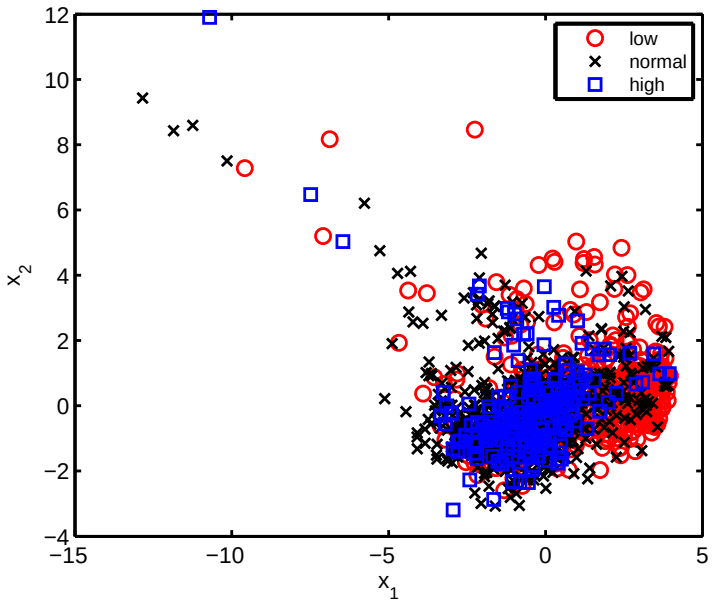

Figure 5.26: Bivariate plot of plant data in terms of linear features obtained using linear discriminant analysis.

Although this may seem reasonable, it should be noted that the three classes are not weighted equally. The 'normal' class contains almost 52% of the data, so that a classifier with no ability to discriminate between the three classes of process conditions could still

score almost 52% by mapping all the data to the class labelled 'normal'. This model failure means that process control could be severely impaired, as the operator would hardly be able to detect abnormal process conditions when they occur and, hence, would also not be able to identify corrective action to be taken when necessary.

Better modelling is possible by making use of kernel methods. The results of a kernel Fisher discriminant analysis using radial basis kernels with different widths are shown in Figures 5.27(a)–(b) as a function of the two kernel-based features F_1 and F_2. Unlike the features L_1 and L_2 extracted with linear discriminant analysis, these features are not explicitly defined in terms of the original variables $\mathbf{x}$. The optimal separation of the three groups was obtained with Gaussian kernels of width $\sigma = 0.75$, found by cross-validating the discriminant model on different training and test data sets. For comparative purposes Table 5.1 also shows the performance levels obtained on the same data set using other proposed extensions to Fisher's LDA (Hastie et al., 2001), as well as nonparametric classification using multilayer perceptron networks and k-nearest neighbors. On this data set, the best performance was obtained using the kernel-based approach, which could classify the data with an overall accuracy of 81%.

Table 5.1: Summary of the performance of different discriminant analysis approaches on data from the liquid-liquid extraction column

Pattern recognition method	Classification error(%)
Fisher's linear discriminant analysis	34
Quadratic discriminant analysis	29
k-nearest neighbor	42
Flexible discriminant analysis (additive spline)	34
Flexible discriminant analysis (multivariate additive spline)	33
Mixture discriminant analysis (3 subclasses per cluster)	33
Multilayer perceptron (single hidden layer, 5 nodes)	28
Kernel nonlinear discriminant analysis	19

The features characterizing normal operational conditions can be used quantitatively to detect faulty conditions by fitting them with confidence limits. Since these features are not consistent with a (bivariate) normal distribution, they are fitted with α-bags instead. The features of the 'normal' class are shown fitted with a 95%-bag (solid line). Any new features mapped outside this α-bag region could be classified with 95% certainty as representative of some process abnormality.

With proper confidence limits, the map shown in Figure 5.27(a) can be used to detect faulty process conditions on the plant as points outside the boundaries of the α-bag of the 'normal' class. However, since the features (F_1 and F_2) do not have physical meaning, the map does not give any indication of corrective action to be taken once abnormal process operation has been established.

Identification of the original plant variables associated with the faulty condition can be done analogous to the way in which contribution plots are generated when principal component analysis or partial least squares models are used (Miller et al., 1998). In this case, a support

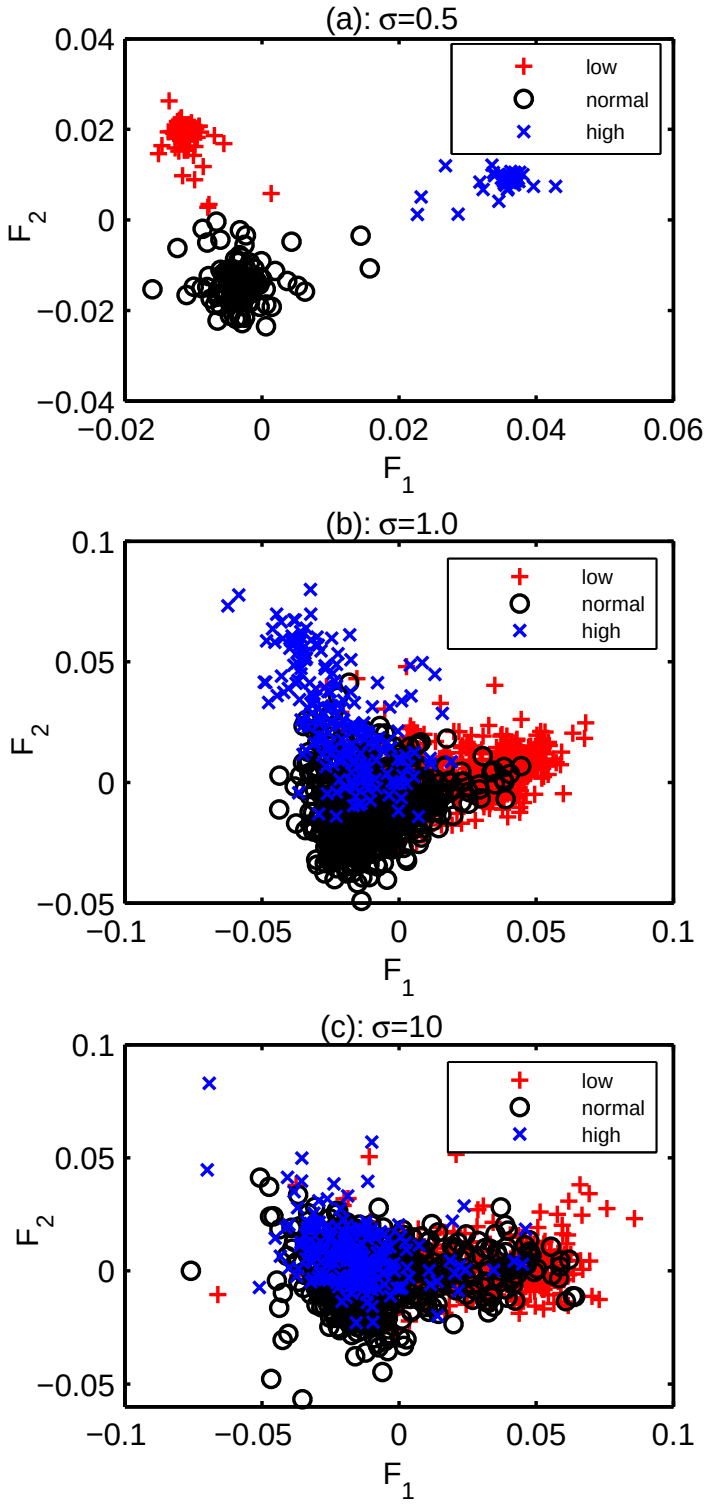

Figure 5.27: Nonlinear features plots for different values of the Gaussian kernel width parameter; (a) $\sigma = 0.5$, (b) $\sigma = 1$, and (c) $\sigma = 10$.

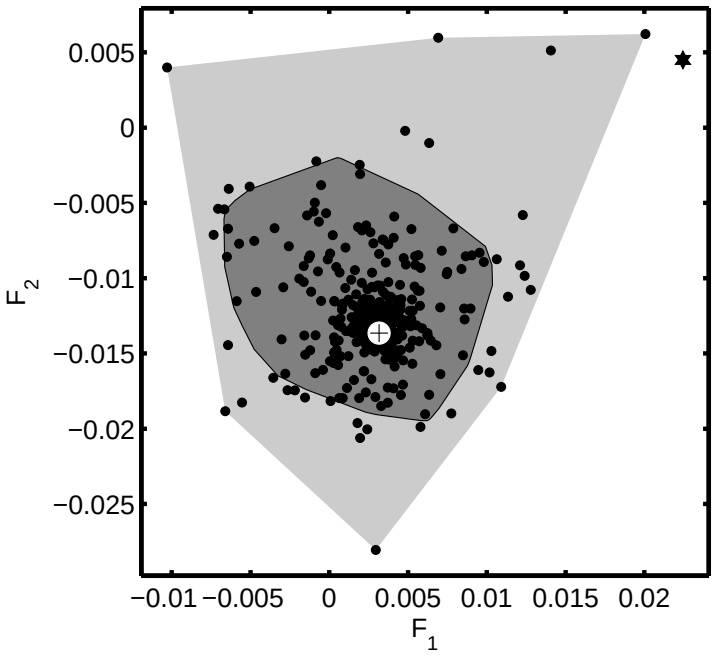

Figure 5.28: 95% bagplot (darker shaded region) of scores of the desired 'norm' class.Outliers are contained in the lighter shaded area, while the mean of the data is indicated by a '+' marker.

vector model was used to relate the scores or features (F_1 and F_2) shown in Figure 5.28 to the original process variables. The input nodes represented the features F_1 and F_2, while the output layer represented the original process variables $\mathbf{x} = (x_1, x_2, \ldots, x_9)$.

New inputs are then mapped to the score space and, if a fault is detected, the scores are projected into the original input space by the model and the residuals between projected and actual coordinates are computed. Having reconstructed the original process variables in this way, it is possible to identify aberrant process variables from the residuals of the reconstructed variables (i.e. the differences between the reconstructed and original variables). Combining results from this analysis and domain expert knowledge eventually assists process operators to rectify incipient faults as they arise.

In this case, Figure 5.29 shows the scores of the plant data, reflecting normal ('norm') operating conditions, as well as two faulty conditions ('low' and 'high') characterized by excessively low and high viscosity indices (the key performance variable of the plant).

Figure 5.30 shows a combined contribution plot when the process moves from point N on the map in Figure 5.29 to point L (i.e. from normal operating conditions to the 'low' fault condition) and from point N to point H (i.e. from normal operating conditions to the 'high' fault condition). The effects of these changes on the absolute values of the scaled residuals of the reconstructed variables are indicated by white bars (N to L) and shaded bars (N to H) in Figure 5.30. All residuals have been scaled with the standard deviation of the scores from normal operating conditions. As can be seen from Figure 5.30, variables x_6 to x_9 (temperature measurements showing increased temperatures) are clearly implicated in the lower viscosity indices of the product. Also note that these changes are not particularly

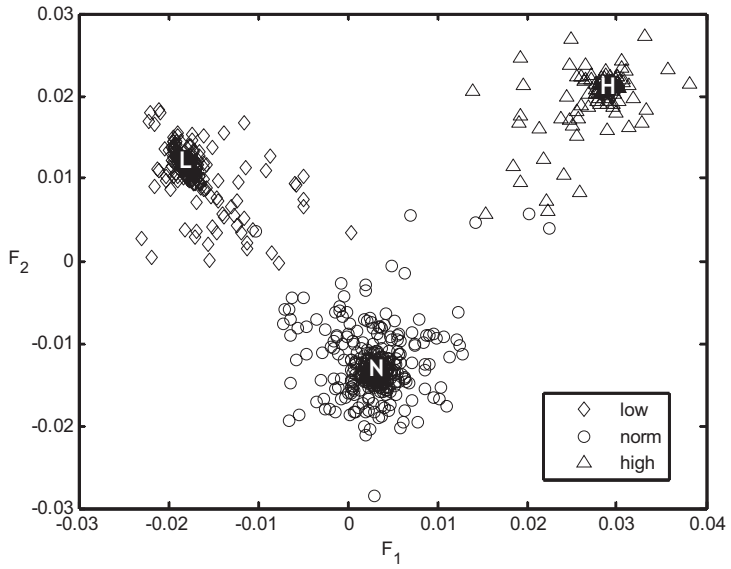

Figure 5.29: A score plot of the process data showing normal operating conditions ('norm') as well as two faulty conditions associated with excessively low ('low') and high ('high') viscosity indices, similar to the ones shown in Figure 5.27

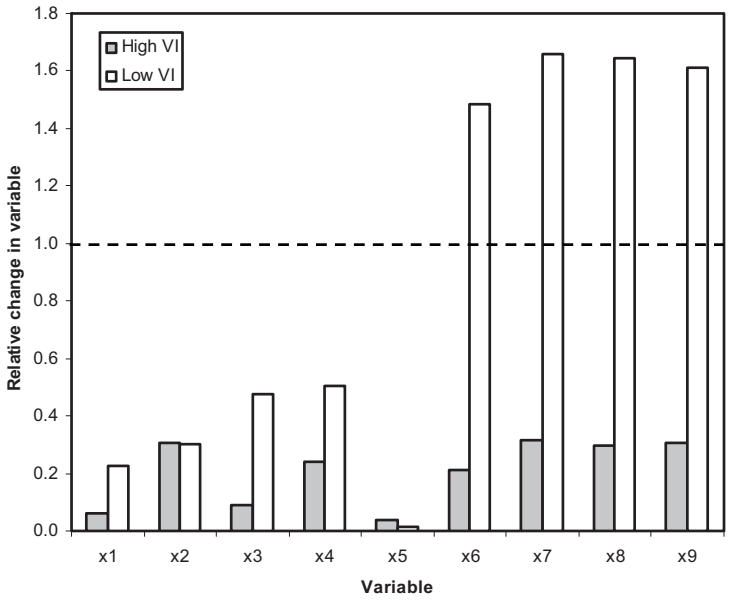

Figure 5.30: Contribution plot showing the variables implicated for inducing change that shift process from normal to excessively high ('High VI') and low ('low') viscosity indices.

pronounced (i.e. less than two standard deviations of the fluctuations in normal operating conditions), hence the poor discrimination of the fault with linear methods, as indicated in Figure 5.26.

In contrast, no individual variables can be identified when the fault 'high' occurs, as indicated

by the shaded bars in Figure 5.30. The difference between the two classes therefore appears to result mostly from differences in the correlation structures of the data from the classes, rather than large changes in individual variables. This is again supported by the data in Figure 5.26, where the two classes are practically indistinguishable from one another in a linear feature space. Unfortunately, under these circumstances correction of the fault through manipulation of one or more of the process variables $(x_1, x_2, \ldots, x_9)$ is not obvious and strategies would probably have to be devised from simulation studies and plant experience.

To summarize, the use of kernel-based methods in monitoring an industrial liquid-liquid extraction improved detection of abnormal process behavior compared to linear methods. Of particular significance in this case study was the observed overall improved performance of kernel-based nonlinear discriminant analysis when compared to other nonlinear approaches. It was demonstrated that, in principle, fault identification via reconstruction was feasible. In the case of an industrial liquid-liquid extraction system, faulty conditions were not necessarily related to large changes in individual process variables, but could also be attributed to changes in the correlations between the process variables. This can lead to other, possibly more complicated control strategies.

5.5 Analysis of the Fault Diagnosis Problem

Residual evaluation for fault interpretation is an important final task in the general fault detection and diagnosis framework discussed in Chapter 2. To completely identify the root cause(s) of a detected fault condition in a process, knowledge of all possible faults associated with the process must be available. This is an ill-posed decision logic problem since it is impossible to have *a priori* information of everything that can possible go wrong on a process. Although in some cases certain fault conditions, particularly those associated with sensors, can be simulated by use of a process model, the design of diagnostic system must ideally include the possibility of novel conditions that have not yet been experienced.

Generally, given known fault conditions associated with a process, including the fault-free or normal operating condition, the fault diagnosis task involves fitting a model between inputs and possible outputs using a training set. The outputs typically take the form of an indicator matrix such that for a process with N known faults, the outputs vector lie in an $\mathbb{R}^{N+1}$ dimensional space, each column representing a specific fault. The extra column in the outputs vector is to accommodate the normal operating condition, say f_0. Each column is assigned a unit entry if the input vector corresponds to that fault, otherwise it is assigned a zero entry. For a problem with N classes, the decision function generator builds a model for each i, for $i = 0, \ldots, N$. When presented with a new instance $\mathbf{x}$ (typically, a symptom or residual vector from preceding residual generation stage), it is input into each model, which then computes an output $f_i(\mathbf{x})$. These outputs are evaluated in the fault decision logic, that is

$$f^*(\mathbf{x}) = G(f_0(\mathbf{x}, f_1(\mathbf{x}), \ldots, f_N(\mathbf{x})) \tag{5.8}$$

where G is the decision logic evaluation function and f^* the most probable offending fault. Essentially, fault diagnosis is a pattern recognition problem. Therefore, pattern recognition

algorithms are typically used to solve the problem. These include distance-based classifiers, multilayer perceptrons, radial basis functions, and support vector machines among other. Figure 5.31 is a schematic illustration of the fault diagnosis problem as discussed above.

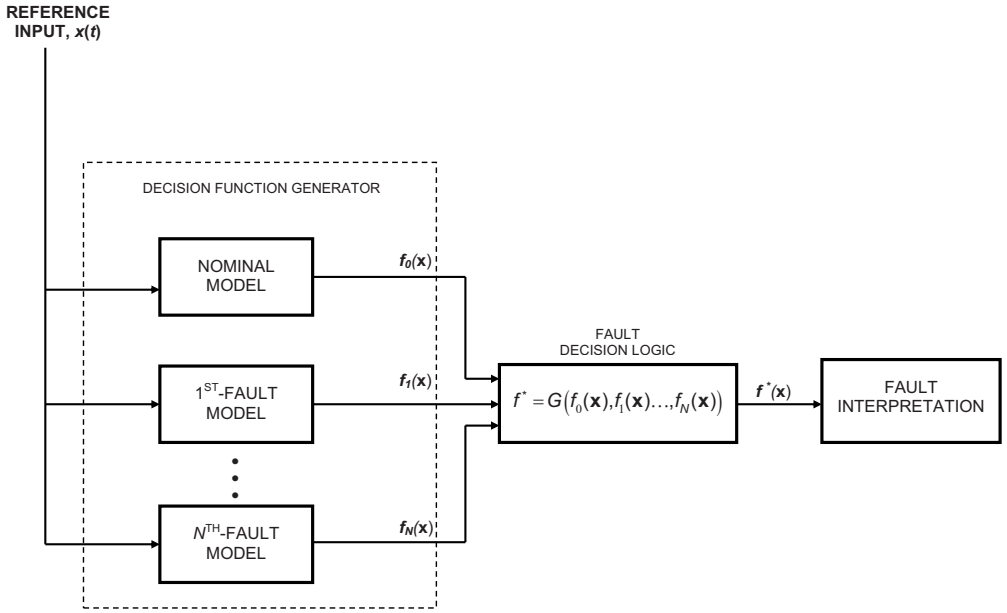

Figure 5.31: Fault diagnosis as a multiclass pattern recognition problem

5.5.1 Fault Diagnosis with Neural Networks

Multilayer perceptron networks are among the most widely used models in solving this problem (Sorsa and Koivo, 1991; Venkatasubramanian and Chan, 1989). In a critical analysis of the fault diagnosis using MLPs, Kramer and Leonard (1990) investigated performance of neural classifiers under non-ideal conditions that reflect conditions encountered in practice. The objective of the study was to identify the problem characteristics that may cause MLPs to perform suboptimally in practice.

It was observed that the tendency of MLPs to extrapolate when a new sample falls outside the training range led to some severe performance problems of the models. Five situations were identified in which extrapolation from the training data was required (Kramer and Leonard, 1990):

1. Small training sets;
2. Changes in parent distributions of the classes occur after training;
3. Corrupted data by, for example, faulty sensors;
4. Appearance of a novel fault class; and,
5. Training the network with synthetic data.

Through a series of investigations, it was concluded that distance-based classifiers should be used instead of MLPs in fault diagnosis problems, because of their greater reliability when dealing with non-representative training data.

5.5.2 Fault Diagnosis Using One-class Classification: An Empirical Analysis

To investigate the use of one-class SVMs in fault diagnosis, the algorithm that uses available fault information as *a priori* knowledge of what the abnormal class looks like, instead of implicitly assuming the novel class is located at the 'origin' in feature space (Equation (3.98) (Schölkopf et al., 2000b) was considered. Similar to Kramer and Leonard (1990), the following generalized fault diagnosis problem was considered;

$$\mathbf{X} = \mathbf{X}_0 + f(\mathbf{p}) + \mathbf{v}, \tag{5.9}$$

where $\mathbf{X}$ is the matrix of measured variables sampled from a static process with a nominal operating steady state $\mathbf{X}_0$, $\mathbf{p} \in \mathbb{R}^{n_p}$ is the fault parameter vector, f is a function incorporating the directional effect of one of the fault parameters on the measurements, and $\mathbf{v}$ is vector of random measurement disturbances. Assuming f is a linear operator $\boldsymbol{\alpha}$, Equation (5.9) simplifies to

$$\mathbf{X} = \mathbf{X}_0 + \boldsymbol{\alpha}\mathbf{p} + \mathbf{v}. \tag{5.10}$$

Fault conditions were classified into groups C_k by defining inequalities on the failure parameters according to

$$C_k : g_k(\mathbf{p}) > 0, \text{ for } k = 1, \ldots, N. \tag{5.11}$$

The model as described is very general and fits many models such as models of continuously stirred tank reactors (CSTRs) in series arrangement and other reactors (see Kramer and Leonard (1990) and references therein).

Let $\mathbf{X}_i \in \mathbb{R}^2$ with $\mathbf{X}_0 = (0, 0)$, the fault $\mathbf{p}$ parameter vector assuming at most a single nonzero element, and the measurement noise $\mathbf{v}$ be Gaussian. Furthermore, the fault classes are defined as:

$$\begin{aligned} \boldsymbol{\alpha} &= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ \text{Normal } (C_0) &: |p_1| < 0.05, |p_2| < 0.05 \\ \text{Fault 1 } (C_1) &: |p_1| > 0.05 \\ \text{Fault 2 } (C_2) &: |p_2| > 0.05 \\ v_i &\sim \mathcal{N}(0, 0.015). \end{aligned} \tag{5.12}$$

A physical interpretation of the fault conditions is as follows: Fault 1 causes both process variables X_1 and X_2 to deviate in the same direction while Fault 2 results in X_1 and X_2 moving in opposite directions. The normal class occupies the intersection region as shown in Figure 5.32.

A generalized one-class SVM model was built for each of the following kernel widths, σ_{RBF} =[0.25 0.50 1.00 2.00 4.00 8.00] and the resulting decision function for each bank of

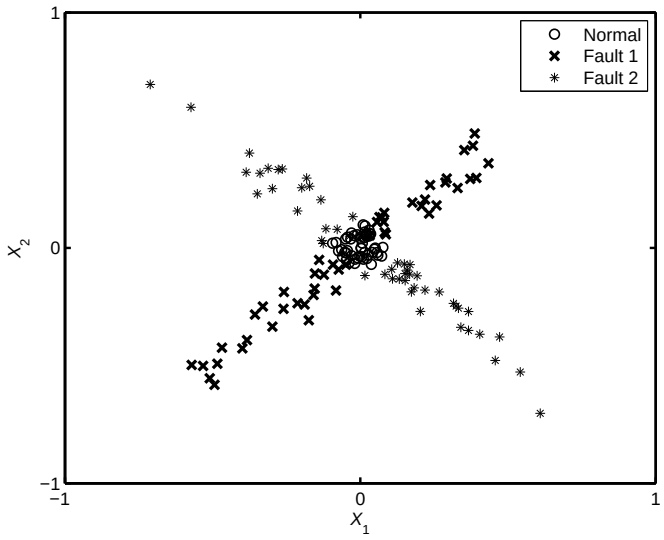

Figure 5.32: 2D generalized fault diagnosis problem of (Kramer and Leonard, 1990).

models is shown in Figure 5.33. For comparative purposes, shown in Figures 5.34 and 5.35 are equivalent decision regions as evaluated by a standard one-class SVM algorithm trained without prior fault information and a binary ν-SVM respectively. It can be noted that unlike generalized one-class SVM none of the other two methods bound the "normal class' in the measurement space correctly. Hence, these other classifiers cannot be expected to perform as well as the generalized one-class SVM from the fault diagnosis perspective, similar to MLPs.

Figure 5.36(a) is a summary of the distribution of errors for 100 independent test sets, each of size 1000 sample points. The classification error is relatively constant within the considered range of the kernel hyperparameter. A scatter plot of a sample of misclassified data points from an independent test shows the general distribution of these errors observed in the simulations, Figure 5.36(b). These errors are distributed in the overlap regions between the three classes and are generally unavoidable. Extrapolation errors were not observed in the majority of the independent test sets. This is in sharp contrast to the findings in Kramer and Leonard (1990) who observed that, because of the arbitrary placement of the decision boundary in an empty region when using multilayer perceptrons, the resultant models had a very high error rate due to extrapolation errors. Hence, it can be concluded that one-class SVMs perform better than multilayer perceptrons in this regard.

To study the robustness of the generalized one-class SVM to small changes in the underlying fault distribution after the classifier has been trained, the changes listed in Table 5.2 were introduced and the results summarized in Figure 5.37. These changes are similar to those investigated by Kramer and Leonard (1990) and were chosen to allow for a comparison with the 1-nearest neighbor classifier that had the best performance in their study.

It can be seen that small changes had little effect on the performance of the pre-trained models. In the case of the one-class SVM model using kernel with width $\sigma = 4$ in Figure 5.37 the following can be seen. An additive change in the sensor bias resulted in moderate

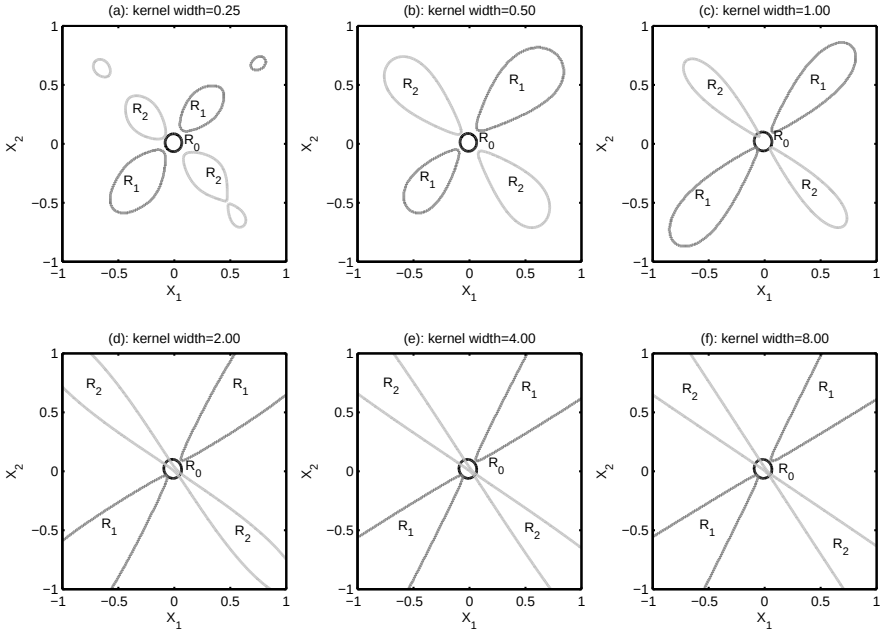

Figure 5.33: Decision regions generated using generalized one-class SVM for different kernel parameters as indicated. The fault model each region represent is indicated by R_0 for the normal region, R_1 for Fault 1 and R_2 for Fault 2. Here, a priori information of the knowledge of the other class was incorporated during training, resulting in the well-defined bounds.

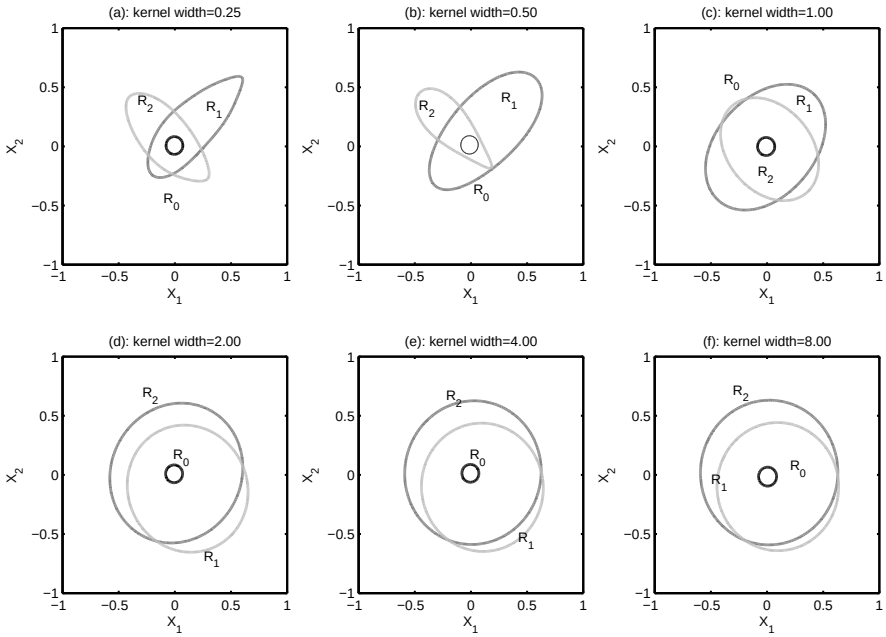

Figure 5.34: Decision regions generated using standard one-class SVM for different kernel parameters as indicated. The fault model each region represents is indicated by R_0 for the normal region, R_1 for Fault 1 and R_2 for Fault 2

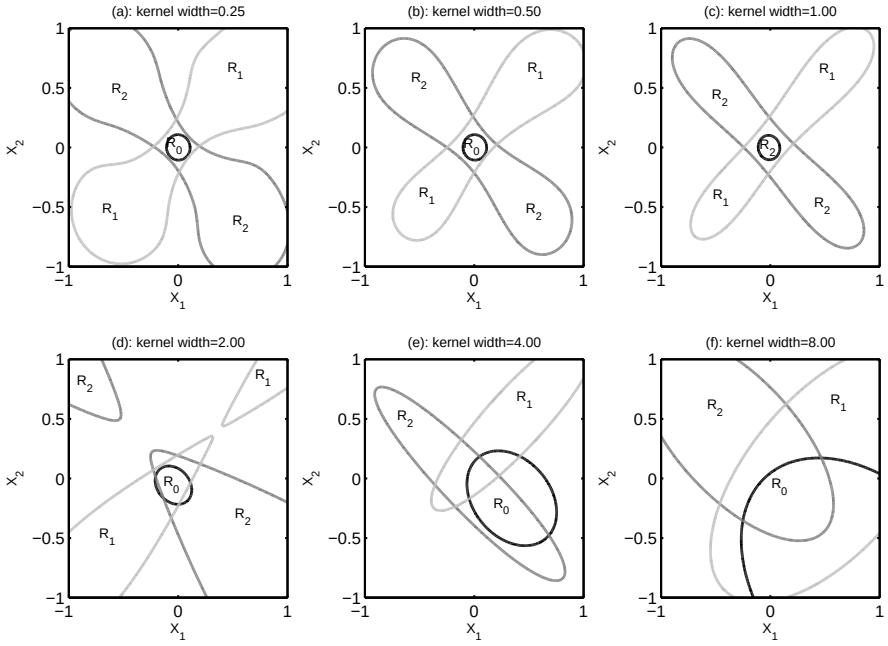

Figure 5.35: Decision regions generated using ν-SVM binary classifiers for different kernel parameters as indicated. The fault model each region represents is indicated by R_0 for the normal region,R_1, Fault 1 and R_2 Fault 2.

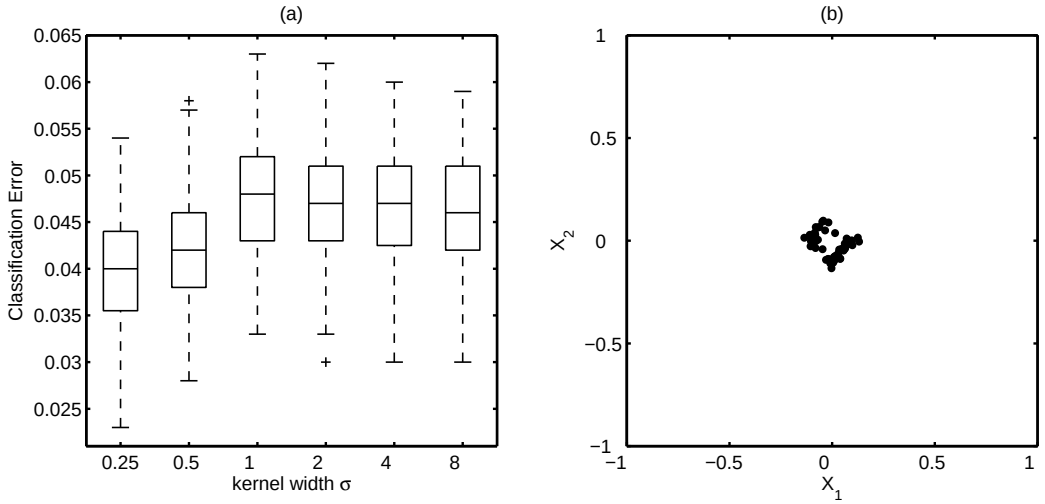

Figure 5.36: (a) Boxplot of classification error for the different kernel widths. 100 test sets were used, each set containing 1000 data points chosen arbitrarily from the parent distribution (b) Typical misclassified patterns. The errors are associated with class overlap since these points lie in the region of intersection of the three models

increases in the classification error, with small and large shifts increasing the error from a nominal 4% to 6% and 4% to 8% respectively. Doubling and tripling the sensor noise uncertainty increased the error rate to 7% and 12% respectively. Small quantitative shifts in the fault directions can arise from process changes. To simulate these changes, small rotational changes in the fault directions were induced to the base case scenario (angle= 45°). A small rotation increased the nominal error rate by 1%, while a large rotation resulted in a 5% change. Finally, changes to the nominal operating point incurred errors of 7% and 9% respectively for the different shifts. With the exception of the rotational changes, the one-class SVM gave marginal to significant improvement in performance compared to the one-nearest neighbour distance-based classifier reported in Kramer and Leonard (1990). Similar results were obtained for kernels with other widths for the same system.

Table 5.2: Robustness Analysis

Error Type	Small Extent Shifts	Large Extent Shifts
Sensor Bias	± 0.025	± 0.05
Sensor Noise	$2 \cdot \mathcal{N}(0, 0.015)$	$3 \cdot \mathcal{N}(0, 0.015)$
Direction	$\pm 7.5°$	$\pm 15°$
Process Drift	$\mathbf{X}_0 = (\pm 0.025, \pm 0.025)$	$\mathbf{X}_0 = (\pm 0.05, \pm 0.05)$

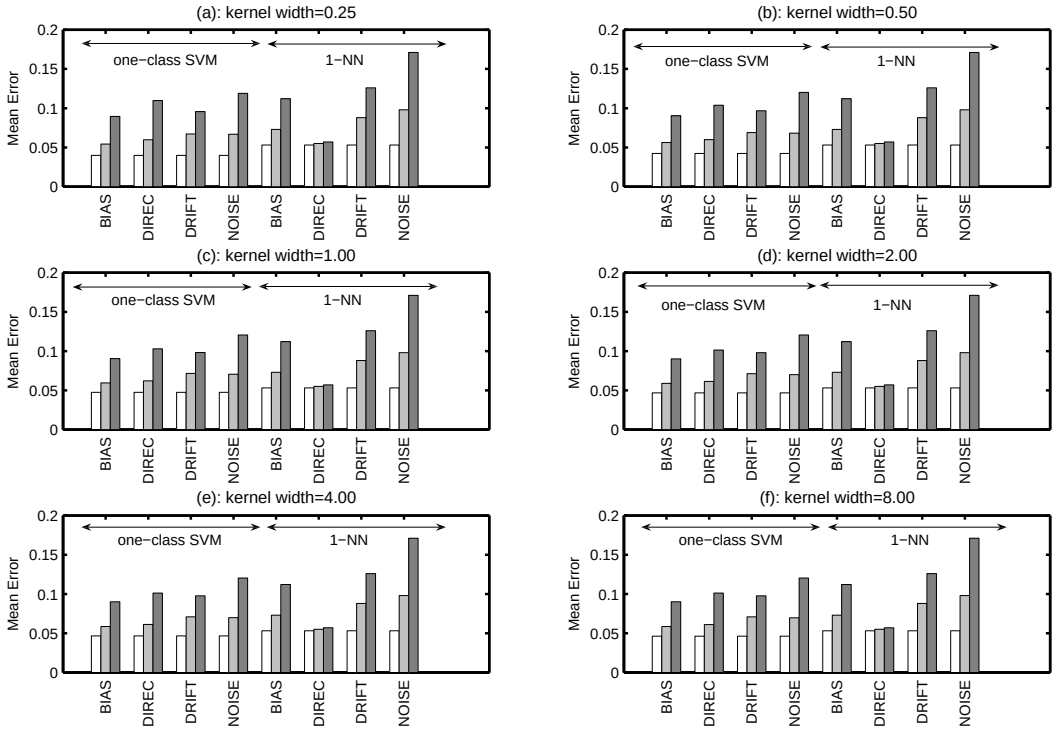

Figure 5.37: Robustness analysis for the generalized one-SVM models. The unshaded block is the base or reference, light gray is for the small extent error, and finally dark gray is for the large extent error.

5.6 Concluding Remarks

In this chapter, a general strategy for fault diagnosis in process plants has been proposed, based on the use of recently proposed kernel methods. The strategy is flexible and can accommodate a wide variety of fault diagnostic elements, such as methods to extract features from the data, various methods to identify faults, once detected, different methods to estimate confidence bounds, etc. The integrated framework presented here is potentially useful in metallurgical processes where fundamental knowledge of the process behavior is inadequate to derive fault diagnostic systems from first principles. An important feature of the proposed methodology is that with the removal of nonlinear structure from the data and subsequent analysis based on the residuals of the data, it is possible to retain one of the main advantages of linear methods, that is the ability to intelligibly relate process faults detected in the feature space to changes in the measured variables.

Although information-rich features, such as those extracted with supervised or unsupervised kernel methods allow better discrimination between normal and faulty process conditions, much of this advantage can be lost if proper confidence limits are not used in conjunction with the features. As was demonstrated in this study, one-class support vector machines can be used to construct nonparametric and non-convex bounds closely fitting the distribution of the supportive data. This allows better fault detection when combined with kernel-based (or other) features that may not adhere to known or homogeneous distributions.

The use of supervised feature extraction methods using nonlinear discriminant analysis indicated an important role the method can play in process monitoring and diagnosis using data-driven methods. This particular method has not received as much attention as the unsupervised version although the benefits of including class information, when available, result in better decision support systems for operations.

Finally, a framework for using one-class SVMs in fault diagnosis was discussed and critically analyzed using a simple two-dimensional system. In spite of its simplicity, the system is representative of a number of industrial reactor systems. Compared to previously proposed nonlinear methods of fault diagnosis using artificial neural networks, the one-class SVM approach was shown to be robust to process changes, insensitive to the influence of data lying in extreme regions, and did not exhibit severe extrapolation errors. Moreover, the one-class approach gave better performance than a one-nearest neighbor classifier, previously proposed as a preferable alternative to artificial neural networks (Kramer and Leonard, 1990).

Chapter 6

Process Optimization with SVMs and Decision Trees

If you optimize everything, you will always be unhappy.

Donald E. Knuth

TO achieve predictable and stable operating conditions, multivariate statistical process control (MSPC) techniques define an "in-control" process model using historical operating data collected under normal operating conditions. The model is subsequently deployed to detect special or abnormal events that may occur during operation using process measurements. Moreover, the reference model is also useful in assisting engineers to focus troubleshooting efforts on reduced subsets of variables in an otherwise high dimensional measurement space. By repeated elimination of root causes of variability, statistical process control methods ensure that the long-term system variability remains bounded. Usually only a few samples that violate control limits from a statistical process control perspective are of interest, while the rest, which may be used to uncover potential improvement opportunities are ignored. However, beyond statistical control an additional step is required to reduce the process variation normally attributed to common causes. To achieve this goal, common and sustained causes not identifiable using MSPC must be interrogated.

In this chapter, a decision support system integrating kernel-based learning methods and inductive decision trees is proposed for identifying process improvement opportunities by reducing common-cause variation. Whereas kernel methods are very effective in capturing discriminative information using a sparse set of instances or exemplars, decision trees are amenable to easy interpretation. The integrated methodology is founded on the basis that success or failure of state-of-the-art approaches are invariably linked to the presence or absence of useful knowledge embedded in the system.

6.1 Background

As alluded to in the introductory chapter, process operations are compelled to continuously challenge current process performance levels and search for improvement opportunities. Data processing and information management have an important role in this regard given the large volumes of data being generated in industrial operations. Pattern recognition and statistical process control methods, in particular, provide a principled framework in the analysis, interpretation, and design of efficient decision support systems based on operating data. The use of univariate and multivariate statistical process control in the systematic search for and subsequent elimination of abnormal process conditions and other assignable causes has been highlighted in Chapters 2 and 5. These methods constrain the time evolution of a process within a state of statistical or predictable control, characterized by bounded variability on a system's behavior. Residual variation is then attributed to unavoidable or common causes which, it is assumed, cannot be eliminated.

Saraiva and Stephanopoulos (1992) proposed a framework for exploring possible improvement opportunities by challenging current performance levels and operating strategies. More specifically, a learning-based improvement strategy that integrates analogical reasoning and symbolic induction instead of planned experimental campaigns was proposed. The rationale behind a learning-based approach is the disruptive nature of experiments in the plant, which are not easily accommodated in day-to-day operations of a plant. In the empirical learning approach, data consistent with a state of statistical control are used. These data are normally of limited interest to both operators and process engineers although it is recognized that the majority of process problems occur in this state. In determining the operating regions and/or operational strategies that yield potential improvement opportunities, a mapping relating process conditions and process trends is conceived using only information generated by the system. Such a mapping ideally must be nonlinear to adequately capture the underlying complex dynamics exhibited by chemical and metallurgical processes.

Despite its appeal, the empirical learning approach is confronted by many issues; more often than not measured process data are discretely sampled (often non uniformly), corrupted with measurement noise, and have an unknown relationship to state variables. In addition, owing to high dimensional measurement spaces encountered in plants, the observed data tend to be sparsely distributed and, hence, problematic when analyzed with established techniques. Also, the underlying process descriptive model may not have a simple closed representation. As discussed in Chapter 3, the empirical risk minimization principle widely used in many learning machines does not guarantee a consistent hypothesis. More specifically, ERM-based algorithms generalize poorly on yet-to-be-seen future data not used in building the model. This is a well-known problem in most learning algorithms, such as neural networks, decision trees, and so on.

In the following sections, a methodology that searches for improvement opportunities through a systematic reduction of process variation by means of SVMs and decision trees is proposed. The approach derives mainly from the work of Saraiva and Stephanopoulos (1992) as well as learning theory considerations discussed in Chapter 3. The key insight is that the problem of process improvements can be reduced to a pattern classification task. Classification or partitioning of the feature space using historical records collected during

"normal" operating periods can be used to detect regions of space in which the process would have a reduced variability.

6.2 Process Improvement Strategies Using Classification of Operating Regions

Saraiva and Stephanopoulos (1992) proposed a decision support system for process optimization based on machine learning approaches, which integrated instance-based and inductive symbolic learning procedures. The system complements statistical tools used in process monitoring and fault detection and diagnosis. Figure 6.1 illustrates the ideas underlying the approach.

The process depicted is assumed to be operating in a state of statistical control. Acceptance of the final product is determined by a suitable quality index, which assigns the products to one of three categories; A, B, and C, of which the desirable products fall into class B. This delineation allows for the definition of decision boundaries between classes A and B, and classes B and C. In the absence of an appropriate fundamental model, an empirical learning approach can be used to define implicit decision rules that partition the operating zones into the designated classes. An essential requirement of the methodology is correct interpretation of the data so as to direct the operator to strategies that offer most promising and interesting hyper-rectangular zones in the decision space for improved process performance.

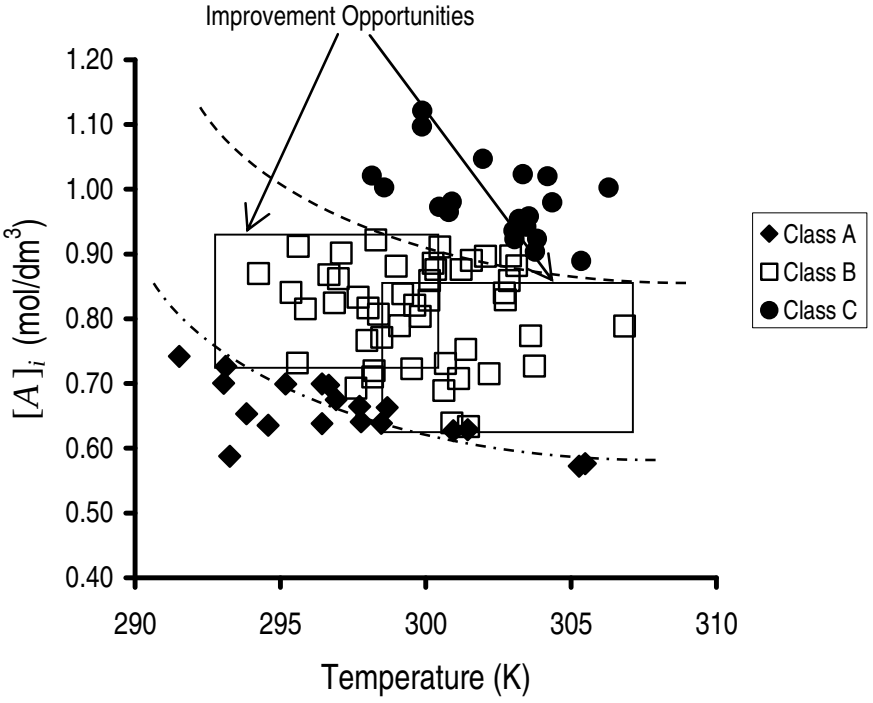

Figure 6.1: Performance improvement through partitioning of regions of space embedding process data points

In their methodology (hereinafter referred to as the SS methodology), Saraiva and Stephanopoulos (1992) used a nonparametric nearest-neighbor classifier to identify the pivotal data points nearer the decision boundary surfaces. Only those points which lie close to the boundary surface and form Tomek links are used to form an active memory of exemplars that are subsequently used to induce a decision tree. As explained later, decision trees are symbolic classifiers that provide a modularized description of the feature space with characteristics that make them suited to the requirements of the methodology. Eventually, the approach suggests changes to current operating strategies and/or design of a reduced set of confirmatory experiments.

However, the SS methodology has certain limitations. Firstly, the use of Tomek links results in piecewise linear classifiers. This may not be appropriate in instances where a decision boundary is correctly described by a nonlinear function. Also, the process engineer has no direct control over the number of training points necessary to induce a decision tree. This is particularly relevant in cases where there are uncertainties in the data. A related issue is that for systems with many variables, larger amounts of training data are required to identify the correct decision hyperplanes. In our experiments (and as indicated in the original work) the methodology has rather slow adaptive properties, making it unsuited for processes with rapidly changing parameters, for example an ore milling plant getting feed from different ore bodies.

The basic elements used in the original development are based on two pattern recognition tools: classification decision trees and memory-based pattern classification. In the proposed approach, support vector machines are used to determine the class boundaries or discriminant functions, while decision trees are retained for their interpretable solutions. An overview of decision trees is discussed next, emphasizing those elements essential in the implementation of a process improvement scheme.

6.3 Inductive Learning Using Decision Trees

Decision trees are hierarchical and structured classifiers that determine classification rules by partitioning the feature space using piecewise hyper-linear decision boundaries. Given a set of process trends or features and corresponding process conditions, a decision tree extracts generalized rules mapping the process features and quality variables. Decision trees discover classification rules by employing a top-down, divide-and-conquer strategy that partitions the given set of objects into progressively smaller subsets in step with the growth of the tree. The derived decision rules are expressed in the form of complexes or conjunctions of conditions amenable to easy interpretation and implementation. Figure 6.2 shows a typical binary decision tree induced using a d-dimensional space of feature vectors. Table 6.1 is the corresponding decision list extracted from the tree.

The decision tree identifies IF-THEN rules using splitting criteria that partition the data set at a node into two maximally homogeneous subsets. Once defined the decision rule can be incorporated into an operational strategy. The splitting rules are based only on the pivotal attributes or variables and, therefore, easy to interpret. This is particularly relevant in situations where data are correlated and therefore contain redundant information, a common characteristic of multivariate data. Construction of a binary decision tree revolves

on a few critical issues; specification of splitting rules at test nodes, termination criteria, and class assignment rules for the terminal nodes. There exist several splitting rules that have been implemented in many algorithms such as ID3, C4.5, and CART (Breiman et al., 1993; Quinlan, 1986, 1990; Utgoff, 1989).

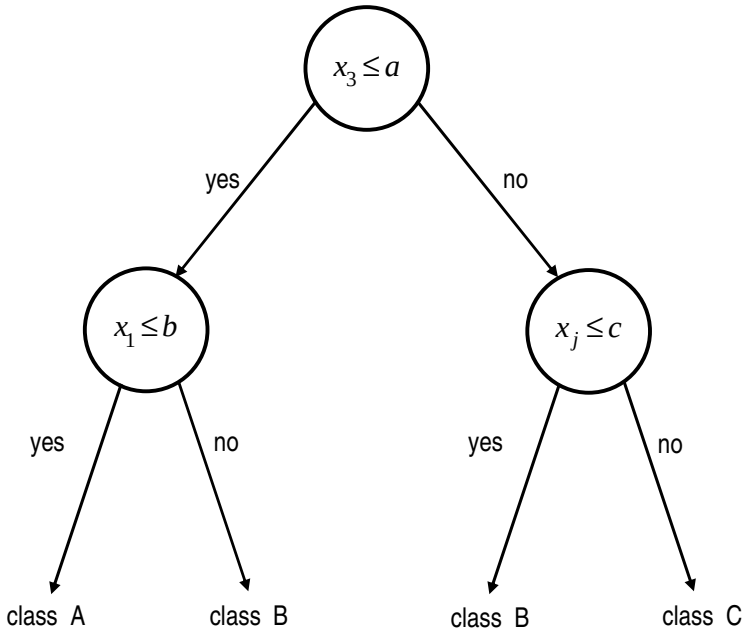

Figure 6.2: Rule extraction using binary decision variables

Table 6.1: Summary of decision rules induced by the classification decision tree of Figure 6.2

Rule	Antecedent(IF)	Consequent(THEN)
1	$x_3 \leq a$ AND $x_1 \leq b$	**x** belongs to class A
2	$x_3 \leq a$ AND $x_1 > b$	**x** belongs to class B
3	$x_3 > a$ AND $x_j \leq c, j \leq d$	**x** belongs to class B
4	$x_3 > a$ AND $x_j > c, j \leq d$	**x** belongs to class C

Although decision trees possess a number of features that are useful in engineering applications (Bakshi and Stephanopoulos, 1994), they have certain structural limitations, which may complicate their use. In particular, the divide-and-conquer strategy does not guarantee an optimal decision tree. It can be difficult to extract intelligible rules from a large and complex tree; and the piecewise linear decision boundaries associated with decision trees may be inadequate for boundaries better defined by nonlinear functions. Fortunately, it is possible to limit these restrictions by integrating decision trees and other empirical learning tools with the desired properties, for example support vector machines.

6.4 Identification of Optimization Opportunities with SVMs

Figure 6.3 illustrates a schematic representation of the proposed continuous process improvement methodology. The overall methodology closely resembles the proposal in Saraiva and Stephanopoulos (1992) except for a few exceptions. To understand the difference, it must be appreciated that central to the original formulation is the pattern recognition module used in building an active memory of cases that lie close to the boundary of the decision hyperplanes. As discussed earlier, the original approach leaves little room for the process engineer to adjust the resulting suggested improvements.

The modifications proposed use support vector classification (SVC) in the selection of the memory of examples used in the induction of trees. As indicated earlier, SVC has different elements that must be decided on; the choice of the kernel, associated kernel parameters, and outlier filtering or detection. This gives more control to the operator/process engineer on the evolution of the suggested process changes. Since the methodology is similar to the original except for the classification task, the discussion that follows focuses on the proposed innovations and associated advantages. The other elements were implemented principally in similar fashion to the original SS methodology.

6.4.1 Description and Illustration of Methodology

The main differences introduced in the proposed approach are with regard to the pattern recognition task and subsequent selection of prototypes or memory of exemplars used in the induction of classification decision trees. Although the focus of the present contribution is on process improvement by exploiting information in process data, the proposed innovations can also be used as separate modules for other related activities, such as fault detection and diagnosis, pattern-based adaptive control, etc. Following Saraiva and Stephanopoulos (1992), a simulated first-order irreversible reaction occurring in a continuous stirred tank system (CSTR) is used to demonstrate the approach, that is

$$\mathrm{A} \rightarrow \mathrm{B}. \tag{6.1}$$

An Arrhenius relationship is assumed for the reaction rate, with activation energy of $99.7\,{}^{\mathrm{kJ}}/_{\mathrm{mol}}$. Furthermore, the feed stream is assumed to contain reactant A only. The concentration of B in the output stream ([B]) is measured at regular intervals as an index for current process performance. The process performance or [B] is a function of four process variables, namely the reactor temperature, T(K), concentration of species A in the feed stream, $[\mathrm{A}]_i$ $({}^{\mathrm{mol}}/_{\mathrm{dm}^3})$, the volumetric flow rate of the feed and output stream, Q $({}^{\mathrm{m}^3}/_{\mathrm{s}})$, and the level of fluid in the reactor, L(m). To allow for visualization of the decision boundaries, the input dimensionality was restricted to only two variables. Hence, the volumetric flow rate and reactor fluid levels were fixed and process operating data were generated using a Monte Carlo simulator for the following Gaussian distributions: $[\mathrm{A}]_i \sim \mathcal{N}(0.8\,{}^{\mathrm{mol}}/_{\mathrm{dm}^3}, 0.1\,{}^{\mathrm{mol}}/_{\mathrm{dm}^3})$, $T \sim \mathcal{N}(300\,\mathrm{K}, 3.5\,\mathrm{K})$. Training and testing sets of sizes 750 and 250 data points respectively were considered in the analysis.

Figure 6.4 is a scatter plot of the process variables using the simulated data. Also shown are the 95% and 99% confidence intervals for the process under the given conditions. These were derived using established MSPC techniques. Since most of the data lie inside

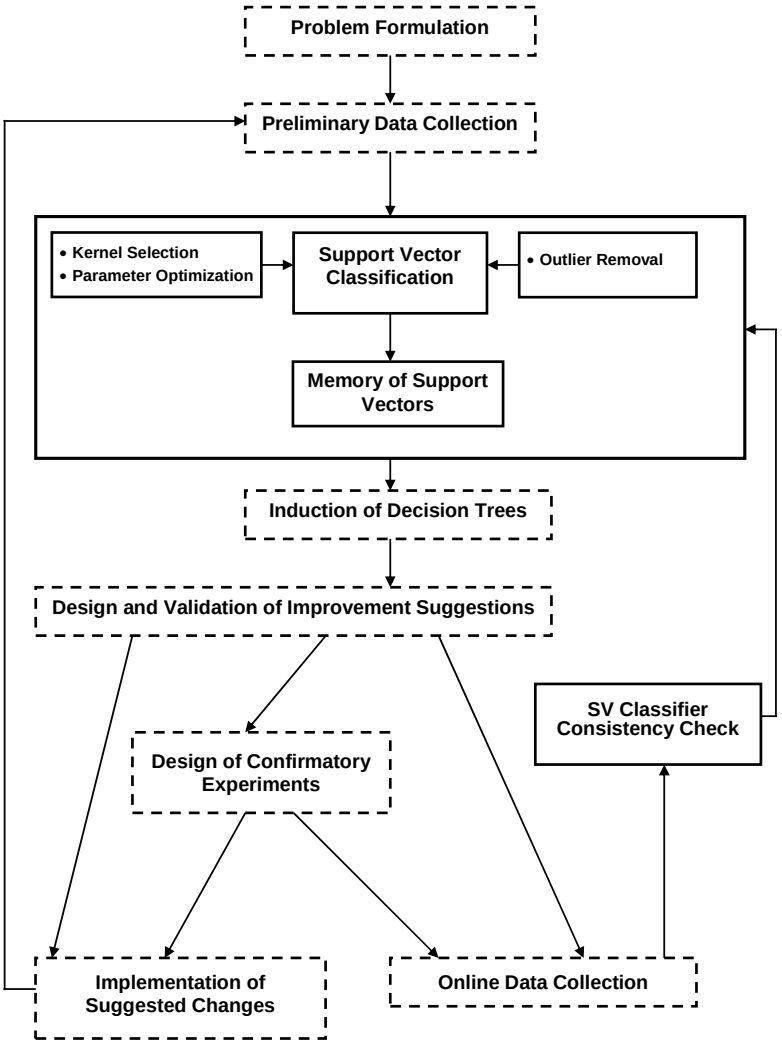

Figure 6.3: A schematic framework for the discovering process improvement opportunities using support vector and decision trees learning algorithms

the control limits, the process trajectory can be assumed to be evolving as dictated by the imposed control strategy. It may not, however, be immediately obvious how the data can be exploited for improvement opportunities, unless other techniques are employed. Below, some of the various functional elements of the proposed modifications are discussed, leading to a descriptive functional representation of the complete methodology.

6.4.2 Problem Formulation

For a real-world process, categorizing products and/or processes into classes using predefined criteria is context dependent. There are several approaches that can be used. A reasonable method is based on the distribution statistics of the quality or performance variables. In the illustrative example introduced above, the current state of the process is assigned into one of three classes, depending on the range in which the concentration of

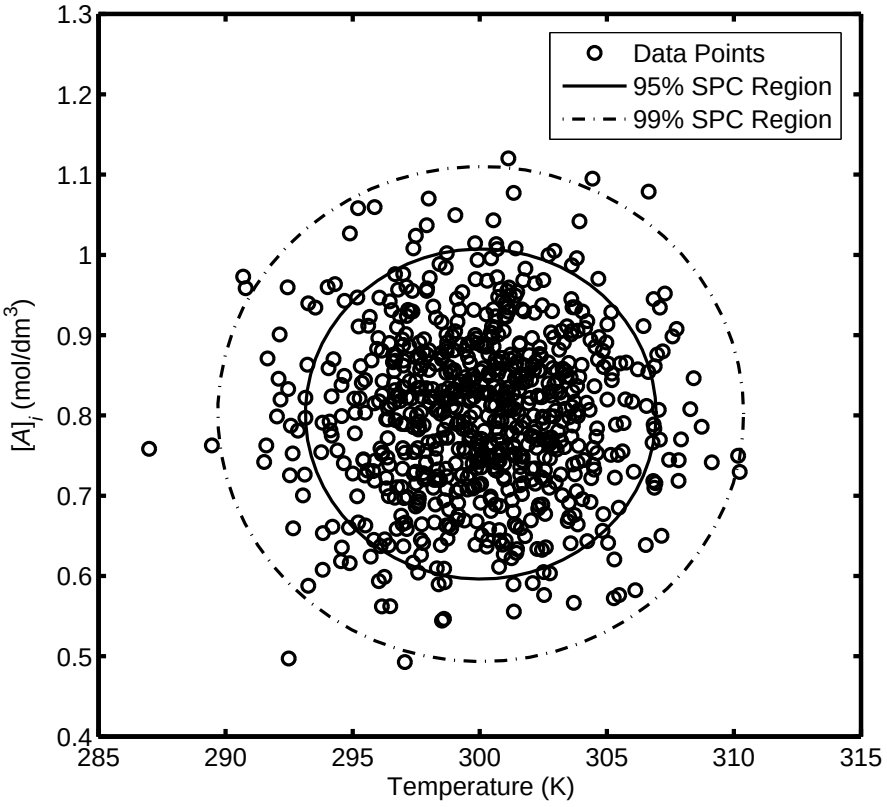

Figure 6.4: Scatter plot of sampled data from the simulated first order CSTR reaction

B lies, i.e.

$$\text{Class A}: \quad [\text{B}] < \mu_{\text{B}} - \sigma_{\text{B}} \tag{6.2}$$

$$\text{Class B}: \quad \mu_{\text{B}} - \sigma_{\text{B}} \leq [\text{B}] \leq \mu_{\text{B}} - \sigma_{\text{B}} \tag{6.3}$$

$$\text{Class C}: \quad [\text{B}] > \mu_{\text{B}} + \sigma_{\text{B}}. \tag{6.4}$$

Classes A, B and C correspond to 'low', 'normal', and 'high' respectively, with μ_{B} and σ_{B} the mean and standard deviation of the quality index. Of interest are instances belonging to the "normal" class. The resulting class separation is shown in Figure 6.5, where also plotted are the true class boundaries between the different classes. For the two-dimensional problem, it can easily be seen that a possible improvement would be to restrict the variation of $[\text{A}]_i$ and T within the ranges $0.7-0.9\,^{\text{mol}}/_{\text{dm}^3}$ and $296-304\,\text{K}$ respectively. The relevance and importance of the strategy becomes even more significant when the feature space is high-dimensional and relationships between the process variables are not as clear-cut.

Using classification decision trees (for inductive symbolic learning), explicit rules for assigning individual cases to the appropriate class can be extracted from the resulting partitioning obtained as shown in Figure 6.6. The corresponding decision tree is illustrated in Figure 6.7, where for simplicity patterns belonging to classes A and C have been grouped into a single set A'. The corresponding rules for assigning a case to class B derived from the

tree are summarized in Table 6.2. Using these rules, the process engineer can search and formulate operational suggestions for process improvement.

Although the procedure is simple, it is difficult to design a supervisory control or similar decision support system for online application, owing to the high computational cost and complex decision rules, particularly for high-dimensional systems. Identification of a sparse and informative representation of the operating data can result in an efficient and robust implementation for reasons mentioned earlier.

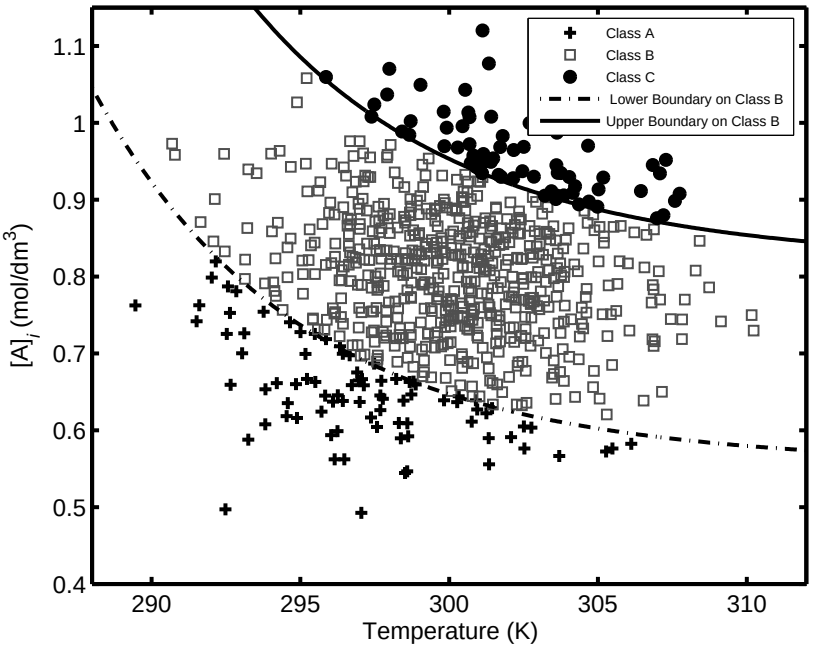

Figure 6.5: Problem formulation of the simulated first-order CSTR system based on linear statistical moments estimates from observed process history.

Table 6.2: Simple decision rules leading to mostly class B for the situation shown in Figures 6.5(b) and 6.7.

$[A]_i$ ($^{mol}/_{dm^3}$)	T(K)	Class
> 0.631	≤ 298.8	B
> 0.826	≤ 293.1	B
≤ 0.905	> 293.1	B
> 0.980	≤ 297.4	B

6.4.3 Identification of Sparse Informative Patterns

Support vector classification (SVC) is a particularly suitable method for identifying a sparse set of informative patterns. As shown in Chapter 3, a support vector classifier is typically expressed as a linear combination of a few of the training patterns (that is, $i : \alpha_i > 0$)

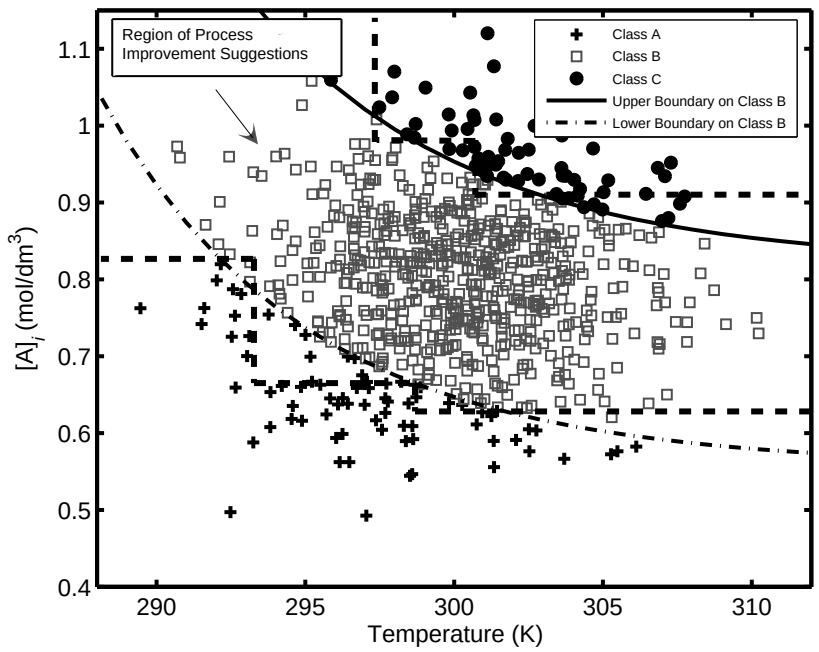

Figure 6.6: Regions of space identified using decision trees for process improvement suggestions. The corresponding decision tree is shown in Figure 6.7.

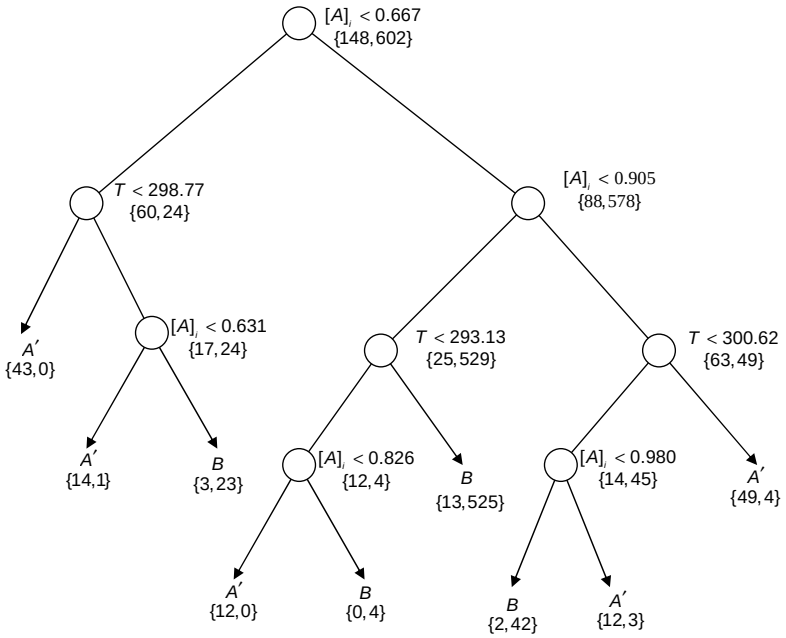

Figure 6.7: Induction of decision tree for the CSTR problem. Here $A' = A \cup C$.

in some feature space, Equation (3.68). Kernel mapping transforms the linear solution in the feature space to a nonlinear function in the original input space. Using the same data introduced earlier, SVC leads to the results shown in Figure 6.8. Here, Gaussian radial

basis kernels of width $\sigma = 0.1$ were used to derive the decision boundaries (solid lines). A regularization constant $C = 10$ was used. Using these support vectors for the induction of decision trees and subsequent refining (Saraiva and Stephanopoulos, 1992) yielded the results shown in Figure 6.9. The corresponding rules for assigning a case to mostly class B are given in Table 6.3. Also, Figure 6.9 shows a typical process improvement region generated by the system and expressed by the following conjunctive rule:

if $0.73 \leq [\mathrm{A}]_i < 0.97$ **then**
    restrict temperature variation within $295 - 300K$
end if

The decision rules induced using support vectors result in similar operational objectives as before even though only fewer samples were used in the inductive learning. This property is of significance in the online application as discussed later.

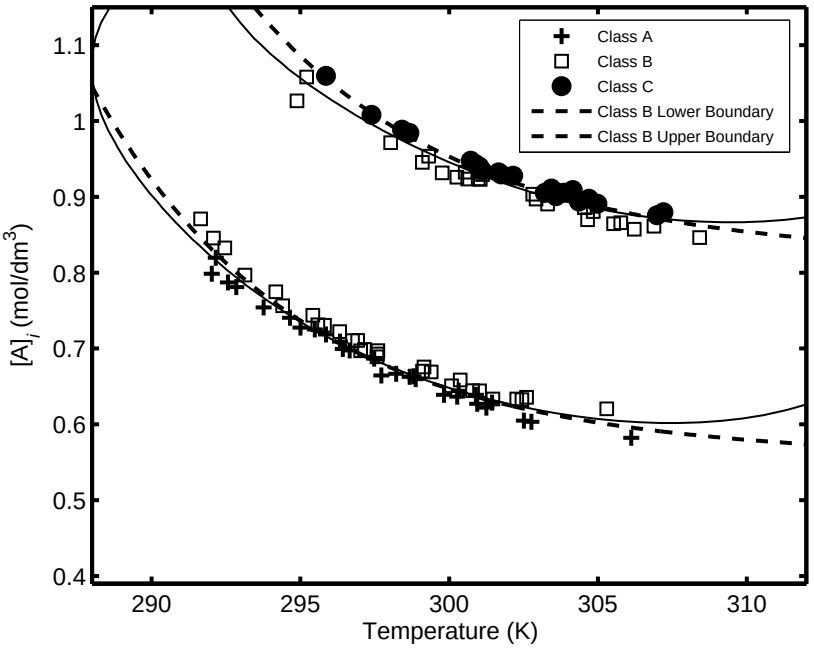

Figure 6.8: Decision boundaries obtained from support vector classification for Class B against Class A examples (top solid line), and Class B against Class C examples (bottom solid line). The respective true boundaries are indicated by the dashed lines.

Table 6.3: Decision list from symbolic induction using support vectors identified in Figure 6.9

$[\mathrm{A}]_i(^{\mathrm{mol}}/_{\mathrm{dm}^3})$	$T(\mathrm{K})$	Class
> 0.826	≤ 295.5	B
> 0.631	> 0.301	B
≤ 0.978	≤ 300.7	B
> 0.978	≤ 295.5	B
≤ 0.929	≤ 0.929	B

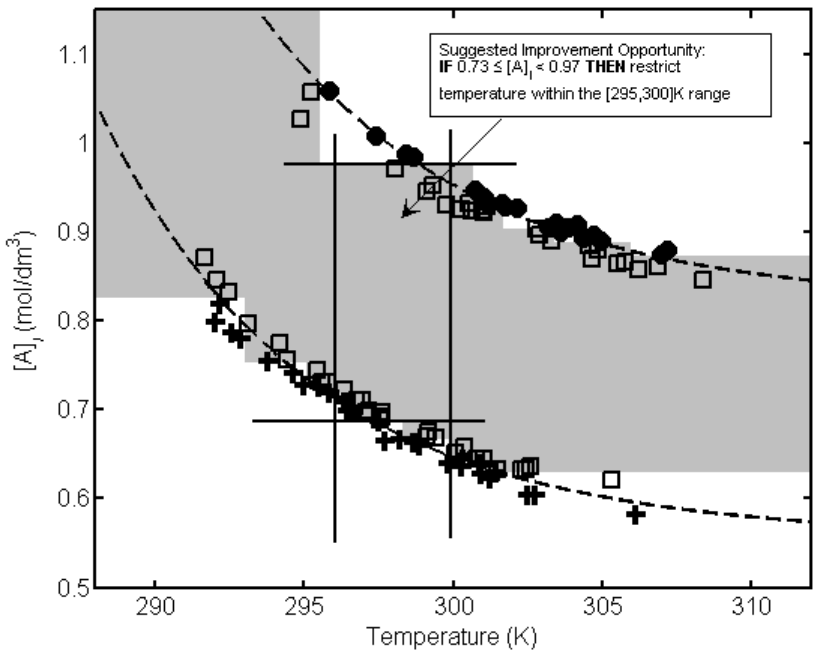

Figure 6.9: Induction and refinement of classification decision tree using support vectors.

6.4.4 Detection and Filtering of Outliers

An important aspect in the proposed method is identification and removal of outliers in the data set. From the discussion on constructing a support vector machine, it will be remembered that it is possible to design a soft margin classifier with slack variables to allow for possible measurement errors through the optimization of Equation (3.66). As $C \to \infty$ the solution obtained minimizes the misclassification error, allowing for as few errors as practically possible. On the other hand, when $C \to 0$ margin maximization dominates and as many errors as possible are then allowed. For support vectors on the margin, the Lagrange multiplier is constrained to $0 < \alpha_i < C$, otherwise $\alpha_i = C$. This feature can be used advantageously to rank support vectors with respect to how important they are in determining the decision surface.

However, use of this property may not be appropriate in outlier detection, as some support vectors, though correctly classified, also have weights identically equal to the regularization constant C. Because the information provided by these near-misses is required, another approach is desirable.

A property which has received little attention in support vector machine applications is the information provided by the values of the slack variables, ξ_i's. Patterns with $|\xi_i| \leq 1$ lie within the margin and are correctly classified and, therefore, convey important information on the decision boundary. However, patterns with $1 < |\xi_i| \leq 2$ are incorrectly classified, although still within the margin. Patterns with $|\xi_i| > 2$ are assigned a wrong class label and, therefore, are immediate targets for elimination. The threshold value of ξ_i for patterns to

be removed can be decided by process experts familiar with the dynamics of the operation, providing an additional control on the decision support system (choice of kernel, kernel parameter, and regularization constant being the others). It should be noted, however, that when there is a shift in the process dynamics, all incoming data will be considered outliers and, therefore removed. Hence, a strategy should be devised to identify the occurrence of such a shift, so that these points are not disregarded. An example of how this can be done is discussed in the section on online implementation of the methodology below.

Figure 6.10(a) shows a typical scenario when there are incorrectly measured patterns in the process database used to initialize the online decision support system for process improvement. After training the support vector classifier, these patterns will be included in the memory of patterns used in the symbolic inductive learning phase. Using the values of the slack variables from the optimization algorithm, these samples can be identified and removed from consideration (Figure 6.10b).

Studies on novelty detection using support vectors have used one-class SVM structures (Schölkopf and Smola, 2002), where the learning problem is formulated in a quantile estimation framework as discussed in Section 3.3.2. It was not considered necessary integrating this approach in the framework. In any case, the task at hand involves processes under statistical control and any abnormal event would be identified in complementary modules.

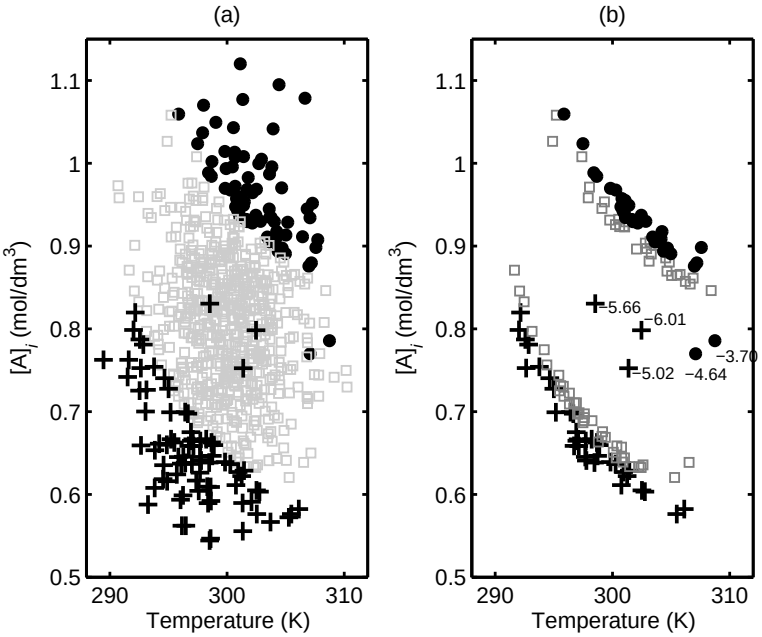

Figure 6.10: Detection of outliers for the simulated CSTR problem using the values of the slack variables after support vector classification. (a) The original data with classes A, B, and C represented respectively by the filled circle, dot and plus sign. (b) The support vectors where those with negative slack variables (xi_i)<2 are designated as outliers

6.4.5 Adaptive Characteristics/Evolution of Memory of Support Vectors

Changes in process conditions, such as changes in water quality, ore type or chemical reagent specifications, may result in process drift inconsistent with prevailing decision boundary definitions. Moreover, it may happen that some of the initially identified support vectors may be inaccurate measurements, which may pass undetected by the outlier filtering step. This is particularly so when these patterns are located at the boundaries of the feature space. As more information becomes available from online data collection, the initial decision boundary surfaces need to be adjusted to reflect an accurate picture of the underlying process behavior. Support vector classification uses all training samples (as a kernel matrix) in finding a solution to the optimization problem. As the number of training samples increases, so do the overhead costs on computation. For batch training, heuristics such as chunking and decomposition, sequential minimization optimization, etc., have been proposed to simplify the problem. Different schemes for online support vector learning have been proposed (Cauwenberghs and Poggio, 2001; Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002). In the methodology proposed, a pseudo-batch online training strategy is used, which is described next.

As each new data pattern is collected, it is classified using the present support vector decision boundaries. An SVC update criterion is defined using the rate of misclassification over a user-specified window interval. Too large a window may result in fewer updates than necessary, while too small a window may result in more updates than necessary. Therefore, the optimal window size should be chosen by the operators conversant with the process, or by trial and error. Another critical index to monitor is the growth rate of the support vectors. For a process under statistical control there should not be large fluctuations in the number of support vectors necessary to define the system. Hence, an unusually large shift in the number of support vectors could indicate novel information not previously explained.

Furthermore, if there is a persistent positive differential in the growth rate of the support vectors, then it may safely be assumed that the underlying process dynamics are changing. In this case, a decision will have to be made on how to update the definition of the decision boundaries. There are two alternatives: (a) redefining the decision boundaries using all previously seen points, or (b) selecting an appropriate subset to re-initialize the support vector machine. Besides being computationally expensive, the first option also leads to retention of class patterns that overlap different classes. Selection of a reduced subset of past operating data remains the principled option, but raises other complicated issues such as the size of the time window and dealing with the situation when a time window does not contain patterns of all classes. It is proposed to use a time window as specified by the user. Also, if a particular class is lacking from the time window, information of that class from the current set of support vectors is retained. From experimental data it was observed that the exclusion of information from all classes resulted in a degenerate decision support system. This degeneracy is related to the method used in building multi-class support vector machines described next.

A one-vs-one method multiclass classification method was used in which one constructs all possible binary classifiers from the n-class data set. Each classifier is trained on only two of the n classes, resulting in a total of $n(n-1)/2$ classifiers. To classify a pattern, these classifiers must be combined, for which various algorithms have been suggested, such

as the Max Wins method, or directed acyclic graph support vector machine (DAGSVM) (Platt et al., 2000). Each method used has its merits and disadvantages and a comparison of some of the approaches based on numerical performance can be found in, for example Hsu and Lin (2002). An efficient implementation of DAGSVM was used in the experiments (Cawley, 2000). To avoid the degeneracy pointed above, the implementation ensures at least one member of each class is available at any given time. For robustness, a higher minimum number can be chosen.

An important consideration is that the support vector classifier update module and the outlier filtering module need inter-linking, as the action of one may negate the action of the other. Filtering is avoided when the update module has been invoked and a process drift has been detected.

Figure 6.11 is a detailed summary of the proposed implementation of the online decision support system for process improvement opportunities. The symbolic induction module is similar to that proposed in Saraiva and Stephanopoulos (1992).

Implementation of the Online Decision Support System for Process Improvement on a Simulated CSTR System

The SVM-based decision support system for process improvement was run online using the previously studied simulated CSTR system in which an irreversible first-order reaction is occurring. The quality variable was selected as the concentration of reactant B in the output stream, classified into one of three classes, 'low', 'normal' and 'high', (or A, B and C) based on the distribution of the quality variable as before. To illustrate the relatively fast adaptive properties of the algorithm, a Monte Carlo simulator was used to generate 23000 points. As in (Saraiva and Stephanopoulos, 1992), the activation energy constant was then changed from $99770^{J}/_{mol}$ to $101430^{J}/_{mol}$ after 10000 time units. The parameter change introduces an upward shift on the class separating boundaries in the input space, assuming the classification scheme remains unchanged.

Figure 6.12(a)-(f) are snapshots of the contents of the support vector set constituting the memory used in the induction of the decision trees. The first 40 points of the simulated data were used to initialize the memory base. A numerical increase in the number of support vectors is observed as more data become available. (However, as a fraction of the training data, a decrease actually occurs). Remarkably, after only about 200 time units the number of support vectors appears to saturate around 80 data patterns. Thereafter, a drastic increase (more than double the saturation level) is observed around 10000 time units as well as an increase in the running error rate. Certainly, such an increase cannot be attributed to measurement errors only. At this point, frequent support vector classifier updates are observed. The decision support system notes that reconfiguration is required and, accordingly, re-initializes the support vector memory by using patterns within the last 50 time units (or other time interval specified by a human expert). It is essential to ensure that all possible classes are defined in the memory base. Hence, in the absence of any one class within the specified interval, all or some support vectors of the missing class from the last decision boundary definition are retained.

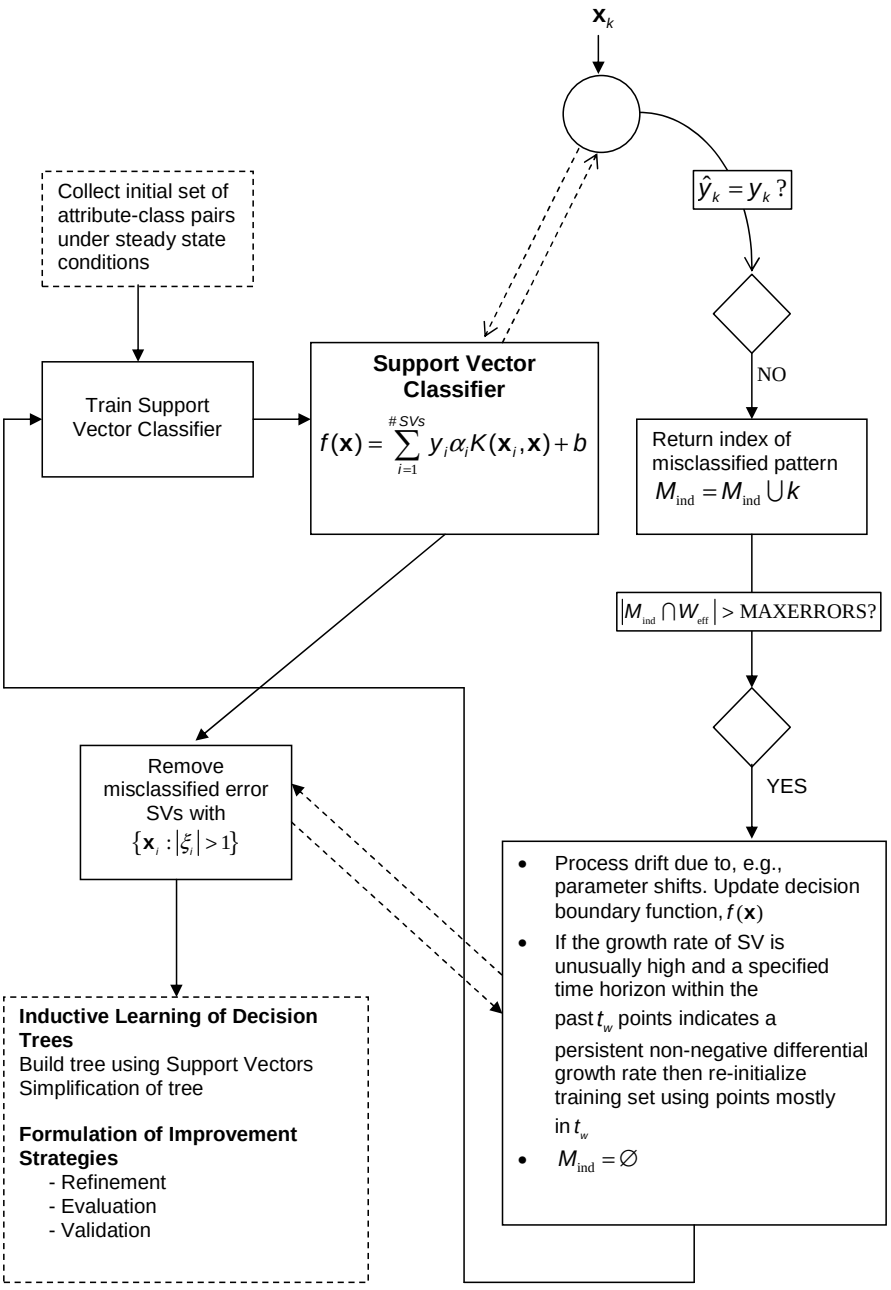

Figure 6.11: Online decision support system for process improvement

This adaptive strategy ensures an immediate change in the nature of the process improvements suggested. For different parameter specifications, it was observed that the system establishes an accurate reflection of the state-of-the-process within 20 time units. Although this is dependent on the system under analysis, it is markedly more rapid than the SS methodology which used a nearest-neighbor scheme on the basis of a Euclidean distance measure, referred to as the DNN-based system in Saraiva and Stephanopoulos (1992). Figure 6.13 shows the corresponding evolution of the active memory of exemplars. Unlike the

proposed SVC-based system, the DNN-based system has a long exemplar-memory saturation time. Moreover, though it is able to detect a change in the underlying process dynamics, a long time lag exists between process changes and system stabilization. This may have a negative impact on a process plant, resulting in a high end-product rejection.

For the choice of parameters used the SVC implementation yields a slightly worse performance compared to the DNN approach, Figures 6.12h and 6.13h. It is also possible to obtain an SVC classifier with improved performance by specifying different model parameters. However, it must be noted that the SVC optimizes the trade-off between the complexity and performance of the model by imposing a maximum bound on the generalization error. Formally, the underlying statistical learning theoretical approach for SVMs seeks to optimize a different criterion (capacity) whereas the empirical risk minimization principle on which the DNN approach is based on minimizes the error on a training set. Hence, at least in principle, the SVM-approach is expected to have better generalization properties.

This has useful implications in the proposed approach. Specifically, variations in input and output measurement errors can be established (e.g. from historical data or sensor specifications) and incorporated into the model parameters, that is, the regularization constant C and kernel hyperparameter(s). As long as the misclassification error is within a certain range one is content that all expected variations have been accounted for.

6.5 Control of Manganese in a Solution Preparation Circuit

Real data from industrial plants occasionally exhibit peculiarities that may require system-specific modifications to the direct implementation of the proposed method. For example, plant data are occasionally collected at sampling rates that do not allow sufficient capture of high frequency process dynamics or, in some cases, sampling may be subject to, for example, biased methods and sensors. In this section implementation of the proposed approach to real data collected from a manganese producing plant is discussed.

As shown in Figure 6.14, the manganese is produced in a sequence of stages including leaching, thickening, and electroplating. Calcined ore containing manganese is fed into a leaching circuit in which dissolution of the metallic species into solution is promoted by the addition of sulphuric acid. Since all metallic impurities have a higher electro-affinity than the manganese, it is important to purify the solution after leaching before electroplating. Impurities are reduced to very low levels by the action of ammonium sulphide during thickening. Finally, the manganese is plated out in the cell house and the residual solution recycled back into the circuit.

A preliminary analysis of the data indicated significant correlation between species concentrations and cell efficiencies, suggesting that solution control in the leaching and thickening stages was critical to efficient plant operation. However, the same data indicated somewhat arbitrary dosage rates for the sulphuric acid, ammonium hydroxide, and alum. It was, therefore, decidedly difficult to develop decision rules based on the species concentration. A possible solution would be development of models for the prediction of the variation of species concentration in the cell house as a function of time-delayed concentrations of species concentrations upstream. Below the variation of manganese concentration in the

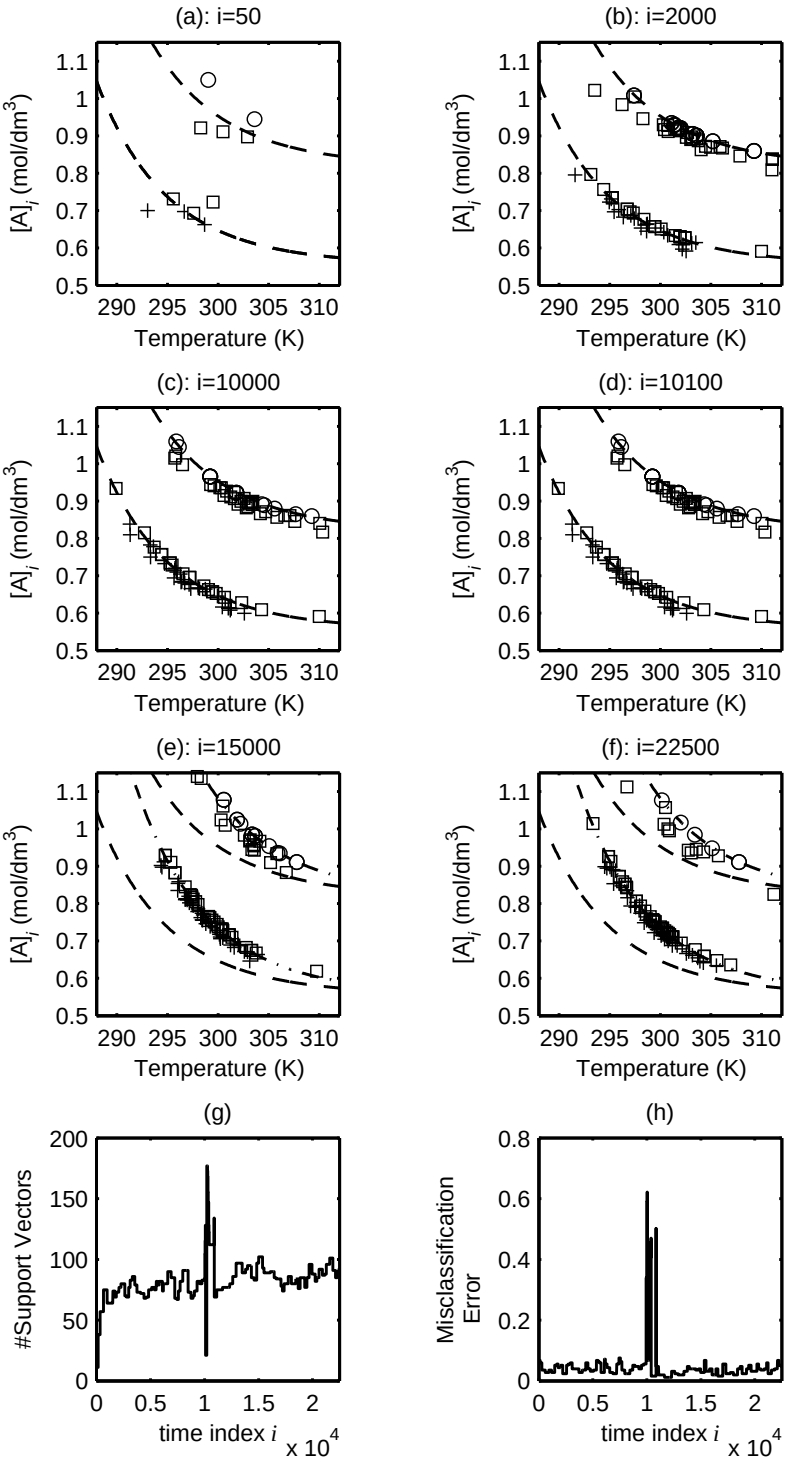

Figure 6.12: Snapshots of the support vectors before (a)–(c) and after (d)–(f) effecting a change in the activation energy at the indicated times. The evolution of the number of support vectors is shown in (g), while that of the misclassification error rate on a test set of the "next" 200 points is illustrated in (h)

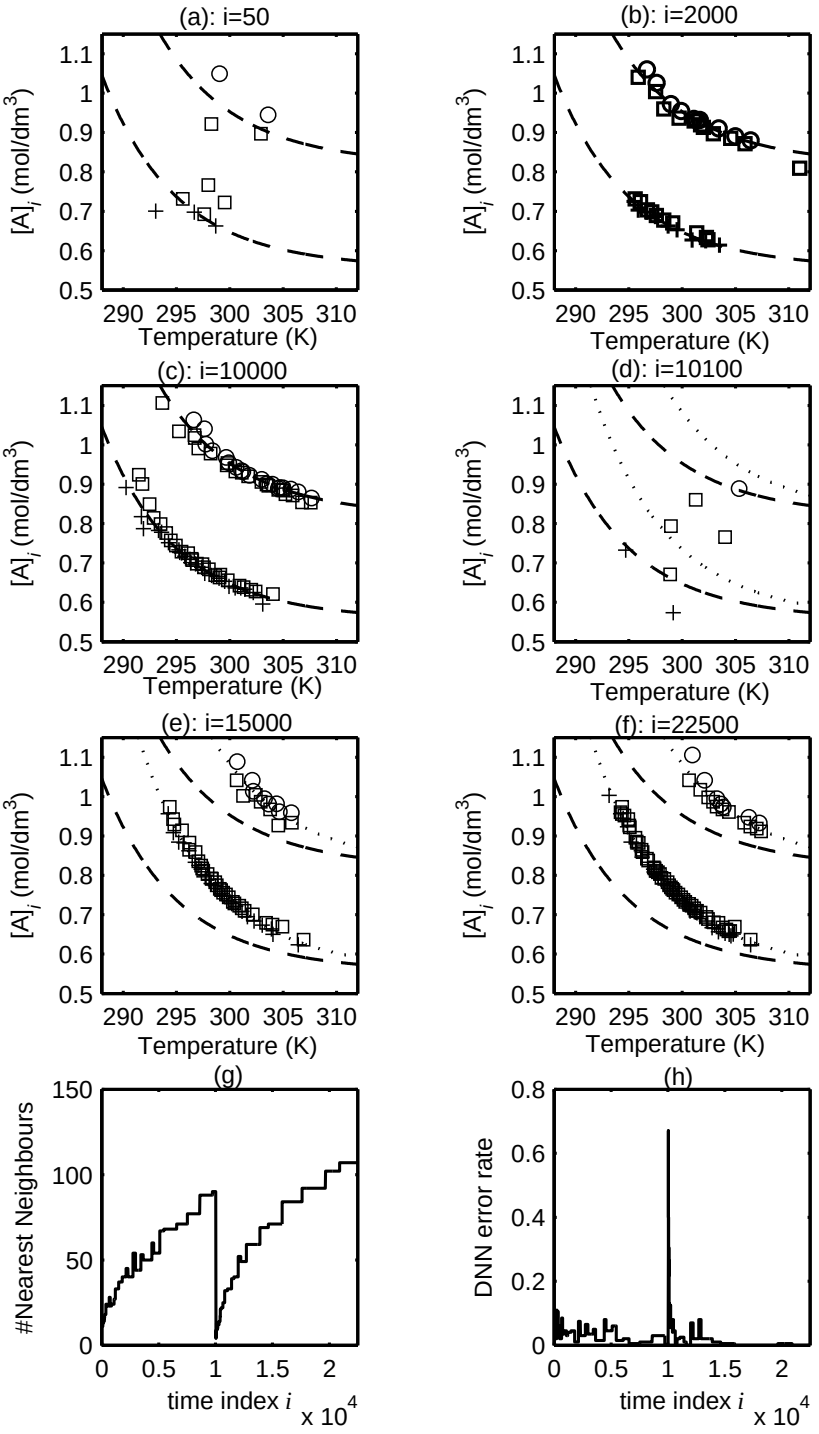

Figure 6.13: Snapshots of the active memory of exemplars in the DNN approach before (a)–(c) and after (d)–(f) effecting a change in the activation energy at the indicated times. The evolution of the number of the nearest neighbors is shown in (g), while that of the misclassification error rate on a test set of the "next" 200 points is illustrated in (h)

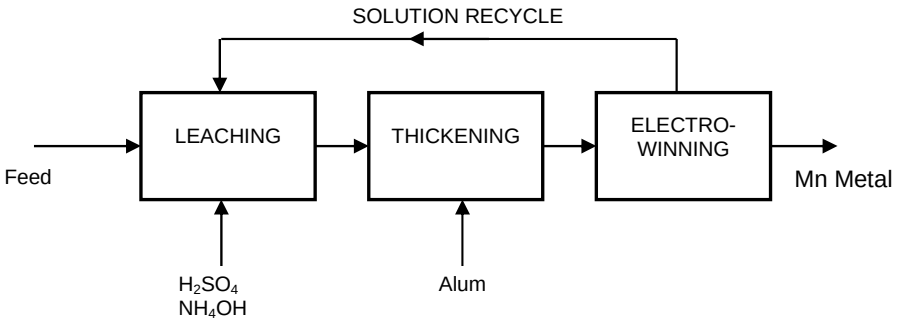

Figure 6.14: Schematic diagram of the manganese metal solution preparation plant

cell house is considered as a function of manganese concentrations in the cell house and leach tanks a unit time earlier:

$$[\mathrm{Mn}(t-1)]_{\mathrm{Cell}} = f([\mathrm{Mn(t)}_{\mathrm{Cell}}], [\mathrm{Mn(t)}_{\mathrm{Leach}}]) \tag{6.5}$$

To search out and formulate possible improvement strategies, $[\mathrm{Mn(t+1)}]_{\mathrm{Cell}}$ is identified as the quality variable, as a function of $[\mathrm{Mn(t)}]_{\mathrm{Cell}}$ and $[\mathrm{Mn(t+1)}]_{\mathrm{Leach}}$. Three classes of the quality variable are specified according to the statistical distribution of the available plant data, as shown in Figure 6.15.

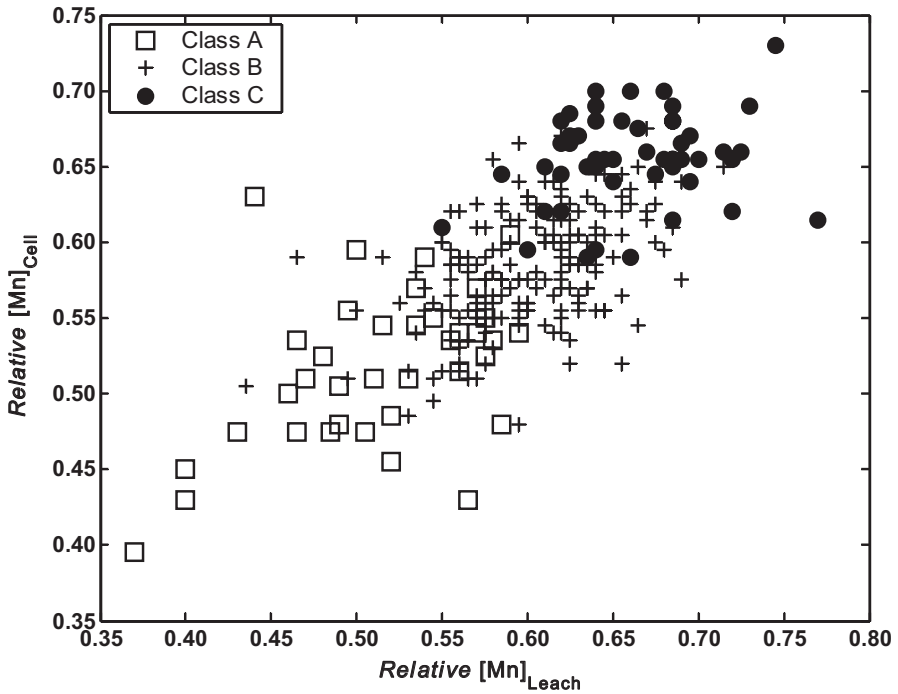

Figure 6.15: Problem formulation for the manganese control problem

As can be seen, there seems to be a linear relationship between the two product variables. A region of desirable species concentration in solution for improved cell efficiencies can be

discerned, although the separation is not as clear-cut. It was found that direct implementation of the proposed improvement framework, though instructive, did not give results suitable for delineating the problem measurement space. This could be attributed to inadequate data for better decision boundary definition. Observing that the crucial data points are those on the boundaries of the desirable class, B in this case, it is proposed to use only those samples in defining a hyper-rectangular zone from which improvement opportunities can be formulated, as indicated in Figure 6.16. Integrating the suggested routes through which the evolution of the process and fundamental chemistry involved in the underlying reactions can be influenced, optimal dosage rates for various reagents can be estimated to improve control of the circuit.

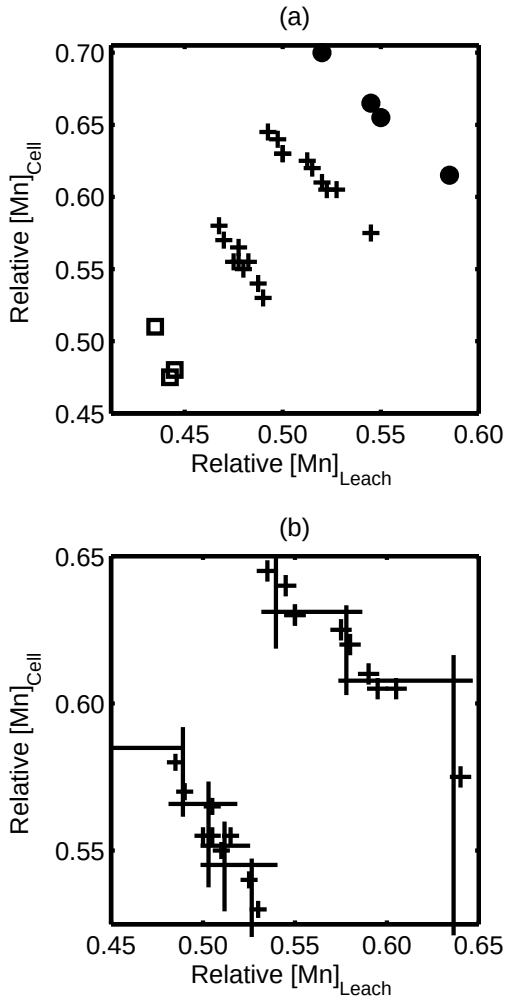

Figure 6.16: Formulating decision improvements for species control in a manganese solution preparation plant: (a) shows the support vectors generated from support vector training. The "+" markers denote the boundaries of the normal operating region, whereas the circles and squares denote the outer boundaries of classes A and C,respectively. (b) The solid lines show the delineation of the operating region supporting opportunities for process improvement.

6.6 Concluding Remarks

In this chapter an innovative modifications to an online methodology for process improvement opportunities previously studied in Saraiva and Stephanopoulos (1992) was proposed. The modifications substitute the central pattern recognition module with one inspired by developments in statistical learning theory, namely support vector classification. It was shown that use of support vector classification in defining the memory of exemplars provides for a number of advantages over the original strategy, including control of the number of data patterns in memory, effective outlier detection and filtering, rapid and flexible adaptive properties, and an ability to handle systems whose decision boundaries are appropriately defined by nonlinear functions.

Integration of support vector classification and a symbolic component (classification decision tree) provides an improvement in online management of product and/or process quality. A salient feature of support vector classifiers is their ability to capture pivotal relationships in a higher-dimensional feature space, which may not be possible in the input space. Thus, though one may measure correlated variables online, implicit mapping into a high-dimensional feature space unmasks these relationships.

Using a simulated CSTR system, various advantages of these modifications were illustrated. A comparison of the online performance of the SVC-based and the original DNN-based systems in showed the superiority of the former. To illustrate the use of the methodology in practical systems, process improvement opportunities were formulated for an industrial manganese extraction plant, where useful approximations could still be formulated, despite sparse and unreliable plant data.

Chapter 7

Conclusions & Recommendations

> As our island of knowledge grows, so does the shore of our ignorance.
>
> John Archibald Wheeler

7.1 Conclusions

A unifying theoretical framework for understanding and developing fault detection, identification, and diagnostic systems is now well-established. The framework allows for integration of various kinds of redundancies to give information that may not be possible to obtain using an analytical model only. In this thesis the diagnosis of process systems has been discussed from a data-driven perspective, which is particularly appealing from a process systems viewpoint given the huge volumes of data being generated on modern process plants. More specifically, the use of a recently introduced family of nonparametric algorithms in developing methods that address some of the current challenges in advanced control systems based on data was considered. These so-called kernel-based algorithms are mainly based on insights from statistical learning theory although they also incorporate ideas from other disciplines such as optimization theory, functional analysis, and neural information processing. They have been applied in various fields where information upgrade of data is also required, for example, bioinformatics, image processing, and signal processing.

The main motivation for using kernel methods is the underlying principled theoretical framework that optimizes a criterion related to the prime objective in learning, that is the margin. Unlike other learning algorithms based on minimizing the training error, optimization of the margin improves the generalization performance of the learned model. This is of particular significance given the uncertainties associated with measured process variables arising from measurement errors and other uncertainties. Moreover, the optimization is related to intrinsic regularities in the observed data and, therefore, there is reason to expect that failure to extract these relations is an artefact of the data and not the learning process. This is in contrast to, for example, multilayer perceptron (MLP) networks whose performance is influenced by the learning process. More specifically, most learning algorithms optimize a non-convex error function that may yield an uneven error surface, resulting in many local minima whereas SVMs optimize a convex error function with a unique global

optimal solution.

The use of the kernel function allows use of flexible expressive models that can capture subtle nonlinearities in high-dimensional data. Because SVMs and other kernel methods learn linear decision functions in high-dimensional feature spaces, only linear statistical complexity is considered in the learning, irrespective of the dimensionality of the data in the measured space as well as sample size. Hence, these methods tend to perform better with high-dimensional data when compared to competing algorithms. Because of the sparseness property of the decision function, computational costs in using SVMs is not related to the number of training patterns used in fitting the model. However, for large databases these informative patterns may still be considerable. In such cases, other methods can give much better performance in terms of computational cost. This is currently a major limitation of these methods, particularly in industrial environments where other methods give statistically comparable performance.

The current study contributed in the following specific areas that are related to monitoring and control of chemical and metallurgical processes.

A nonlinear extension of the singular spectrum analysis using kernel methods was proposed with capacity to significantly reduce energy scatter compared to existing nonlinear approaches when the underlying signal has a dominant harmonic component. Using a simulated example, the method was shown to have promising applications in data rectification, gross error detection and multiscale analysis. Furthermore, an improved (nonlinear) method for classification and characterization of time series data, called kernel-based Monte Carlo singular spectrum analysis, was proposed and evaluated on benchmark systems. Based on the evaluations, the proposed nonlinear variant displayed better performance compared to equivalent linear formulations. However, for some values of the kernel parameter the method did not perform as well. Hence, choosing an appropriate value for the parameter is important in practical applications of the statistical tests. In addition, no physical meaning could be associated with the computed discriminating statistics as is the case in linear formulations. Further work will be directed in addressing these shortcomings.

With respect to multivariate statistical process monitoring, an improved nonparametric confidence bound using one-class support vector machine (SVM) was proposed. The bound is particularly useful in graphical process monitoring charts. Compared to the use of confidence limits based on hypothesized statistical distribution, one-class SVMs are data-based and, therefore, capable of fitting a bound consistent with the observed data. This is important in minimizing error rates from incorrectly detecting a fault (Type I error) as well as incorrectly accepting faulty conditions (Type II error) as often happens with non-normal distributed data. Unlike density estimation methods, bootstrap re-sampling is not required to determine the threshold of the bound. Instead, the threshold is specified by a user-specified parameter ν that has an intuitive meaning - it specifies the rate of Type I error acceptable for the process. The algorithm finds a quantile region corresponding to the bound and specified kernel. Other modifications are possible such as including information on what kind of abnormalities to expect.

A new method based on residual analysis was introduced based on the use of kernel principal component analysis for feature extraction in high dimensional space. Subsequently, the deterministic features observed are removed from the original input space data. Conven-

tional MSPC methods are then applied on the residuals. An advantage of this approach is that it is then possible to attach physical meaning to the resulting model when compared to the case of working in a high dimensional feature implicitly defined by the kernel but, unfortunately, not accessible in the physical sense.

The use of supervised feature extraction methods for process monitoring and diagnosis was explored by means of nonlinear discriminant analysis. Unlike PCA, discriminant analysis has not been as widely used in diagnosis of process systems although it may be expected to perform better since class information is available, as illustrated in the thesis.

A framework for using one-class SVMs in fault diagnosis was introduced and critically analyzed using a simple $\mathbb{R}^2$ system. Despite its simplicity, the 2D system is representative of a number of industrial reactor systems. Compared to previously proposed nonlinear methods of fault diagnosis using multilayer perceptrons, the new approach displayed some desirable properties such as robustness to process changes, insensitivity to influence of data lying in extreme regions, less arbitrary placing of the data boundary in regions devoid of data, and negligible extrapolation errors, with the majority of the errors resulting from class overlap in the intersection of the different regions. Moreover, the one-class SVM had better performance compared with the nearest neighbor distance-based classier. Therefore, it can be considered as a benchmarking model for future diagnostic systems.

A decision support system for process optimization integrating support vector machines and classification trees was proposed. The optimization is aimed at reducing common cause variation normally assumed unavoidable in multivariate statistical process control. In the proposed framework, a support vector machine's capability to learn complex decision boundaries using a few informative patterns is incorporated with the interpretable inductive decision trees. Moreover, a number of properties relevant to fault detection and diagnosis as well were implicitly defined in the model. For example, the adaptive properties of the methodology allowed for other tasks such as detecting process shifts due to parameter changes, as well as outlier filtering.

7.2 Future Investigation

As argued in the thesis, support vector learning and kernel methods can perform better in extracting underlying regularity in data compared to other learning algorithms. Furthermore, they also provide a unifying framework for the theoretical analysis of other algorithms. Despite these advantages, a number of issues still need further consideration in practical application. Firstly, while the computational cost of kernel methods is independent of the dimensionality of the data, it is affected by the sample size. The size of kernel matrix, which encodes all the information in the data, is defined by the size of sample under scrutiny. Very large samples sizes may not fit in the memory of most standard personal computers. A number of methods have been proposed to minimize the effect of large sample sizes, for example stochastic gradient descent methods for online methods (Kivinen et al., 2004), sequential minimal optimization (Platt, 1998), sparse greedy matrix approximation (Smola and Schökopf, 2000). Schölkopf and Smola (2002) discuss these and other methods as well as provide guidelines on choosing the appropriate optimization method for a given problem.

While these propositions are useful, large scale implementations may need to rely on some method of multicore processing using clusters of computers.

Another open area of research is automating the methods for ease of use with inexperienced operators. The main challenge is in selecting or designing the appropriate kernel and optimizing the hyperparameters such as the Gaussian kernel width σ or the regularization term C that are not automatically tuned during training. A promising approach is the support kernel method (SKM) in which a number of pre-specified kernels are used in the training, with their weights tuned automatically (Bach et al., 2004). These has appealing properties for analysis of data with different time-frequency localization since a number of kernels can be used instead of only one. The use of multiscale kernels and other combinations of kernels can improve capture of regularity in data.

While classification and unsupervised learning were only considered, there's also room for investigating use of regression methods inspired by support vector machines, particularly for system identification.

7.3 Publications

Below is a list of publications arising from the study described in the thesis.

Journal Publications

1. Jemwa, G.T. and Aldrich, C. Improving process operations using support vector machines and decision trees. *American Institution of Chemical Engineering Journal*, 51(2):526–543, 2005.
2. Jemwa, G.T. and Aldrich, C. Monitoring of an industrial liquid-liquid extraction system with kernel-based methods. *Hydrometallurgy*, 78:41–51, 2005.
3. Jemwa, G.T. and Aldrich, C. Classification of process dynamics with Monte Carlo singular spectrum analysis. *Computers and Chemical Engineering*, 30(5):816-831, 2006.
4. Jemwa, G.T. and Aldrich, C. Kernel-based fault diagnosis on mineral processing plants. *Minerals Engineering*, 19(11):1149-1162, 2006.

Refereed Conference Proceedings

1. Jemwa, G.T. and Aldrich, C. Inductive learning with classification and regression trees and support vector machines. In: E. Boje and J. Tapson (editors), *Proceedings of the First African Control Conference (AFCON)*, Cape Town, South Africa, 2003:461-466.
2. Jemwa, G.T. and Aldrich, C. Discovering process improvement opportunities with support vector machines and decision trees. In: European Symposium on Computer-Aided Process Engineering-14 (ESCAPE-14) 37th European Symposium of the Working Party on Computer-Aided Process Engineering, ELSEVIER, 2004

3. Jemwa, G.T. and Aldrich, C. Fault Diagnosis in Metallurgical Process Systems with Support Vector Machines. In: Computational Analysis in Hydrometallurgy, 35th Annual Hydrometallurgy Meeting, 2005:61-70.

Appendices

Appendix A

Fault Detection and Diagnosis Terminology

Fault A deviation of at least one characteristic property or parameter of the system from the acceptable or expected condition.

Disturbance An unknown (and uncontrolled) input acting on a system.

Residual A fault indicator, based on a deviation between measurements and model predictions.

Symptom A change in an observable quantity from normal behavior.

Fault detection Determination of the faults present in a system and the time of the detection.

Fault isolation Determination of the kind, location and time of detection of a fault performed after fault detection.

Fault identification Determination of the size and time-variant behavior of a fault performed after fault isolation.

Fault diagnosis Determination of the kind, magnitude, location and time of detection of a fault performed after fault detection. It is constituted of fault isolation and fault identification steps and follows fault detection.

Monitoring A continuous real-time task of recognizing anomalies in the behavior of a dynamic system and identifying faults.

Supervision Monitoring a physical system and taking appropriate actions to maintain the operation in the event of a fault occurring.

Quantitative model Use of a set of static and dynamic relations among system variables and parameters in order to describe a system's behavior in quantitative mathematical terms.

Qualitative model Use of a set of static and dynamic relations among system variables and parameters in order to describe a system's behavior as in terms of causalities or IF-THEN rules.

Diagnostic model A set of static and dynamic relations that link specific input variables - the symptoms - to specific output variables - the faults.

Appendix B

MATLAB Software Codes

B.1 Support Vector Classification

```
% A demonstration of soft margin SVM binary classification
% problem using a 2D toy data set (Figure 3.6)
% This script file depends on the following m-files:
% svtutor_ctrain.m, svtutor_ctest.m, svtutor_kernel.m
% which implement the basic SVM ideas in Chapter 3



kernels = {'linear','poly','rbf','rbf'};
kerparams = [1 2 2 5];
regC = [10 10 10 10];
sym = {'','deg','\sigma','\sigma'};

% create data
randn('seed',100);
m=30; d=1; s=2;
x1=[randn(m,1)*s-d randn(m,1)*s-d];
x2=[randn(m,1)*s+d randn(m,1)*s+d];
d=data([x1;x2],[ones(m,1); -ones(m,1)]);
w=[-8 8 -8  8];
gridsz=250;
xrange=w(1):(w(2)-w(1))/(gridsz-1):w(2);
yrange=w(3):(w(4)-w(3))/(gridsz-1):w(4);
[xs ys] = meshgrid(xrange,yrange);
Xt = [xs(:) ys(:)];
X = [x1;x2];
Y = [ones(m,1); -ones(m,1)];
figure; subplot(2,2,1);
for k=1:length(kernels),
```

```
    model = svtutor_ctrain(X,Y,'kernel',kernels{k},...
     'kerparam',kerparams(k),'C',regC(k));
    Yt = svtutor_ctest(model,Xt);
    subplot(2,2,k); hold on;
    plot(X(Y==1,1),X(Y==1,2),'ko');
    plot(X(Y~=1,1),X(Y~=1,2),'+');
    contour(xrange,yrange,reshape(Yt,size(xs)),[0 0],'k')
    contour(xrange,yrange,reshape(Yt,size(xs)),[1 1],'k:')
    contour(xrange,yrange,reshape(Yt,size(xs)),[-1 -1],'k:')
    set(gca,'YTick',[-5 0 5],'XTick',[-5 0 5],'box','on',...
     'linewidth',1.2)
    axis(w); axis square
    if k==1,
        title(sprintf('%s kernel',kernels{k}));
    else
         title(sprintf('%s kernel, %s=%d',...
          kernels{k},sym{k},kerparams(k)));
    end
end

%-------------------------------------------------------------------
%                              function m-files
%-------------------------------------------------------------------

function model = svtutor_ctrain(X,y,varargin)

% function svtutor_ctrain(X,y,model)
%           Train a binary Support Vector Classifer (SVC) with using a
%           kernel kfun with parameters pars
%INPUTS:
%               X - data matrix
%               y - class labels {-1,+1}
%               kerfunction - kernel function to use
% ('linear' [default], 'poly', 'rbf')
%               kerparam - kernel hyperparameters associated with kfun
%               C      - misclassification penalty
% OUTPUTS:
% DEPENDENCIES
%
%           quadprog, optimset (MATLAB OPTIMZATION tbx)
%           svtutor_kernel
%
% REMARKS:
%               Only useful for demonstrations purposes for sample sizes
%               less than 500.
```

```
%
% Author: GT JEMWA, 2005
% Last Revision: 4 Feb, 2007

% define defaults
C=10^6; % default misclassification penalty
kerfunction = 'linear';
kerparam = 1; % default kernel parameter
sv_cutoff = 1e-3; % sv_cutoff
maxIter=10000; % maximum number of iterations during optimization
ridge = 1e-10; % factor to avoid ill-conditioned kernel matrices

% input argument check
if nargin<2,
    disp('Not enough input arguments.')
    disp('Type ''help svtutor_ctrain'' for more info')
    return
end

if ~isempty(varargin)
    if mod(length(varargin),2)~=0,
        disp('Additional input arguments must be paired.')
        disp('Type ''help svtutor_ctrain'' for more info')
        return
    else
        for i=1:length(varargin),
            if strcmpi(varargin{i},'kernel'),
                kerfunction=varargin{i+1};
            end
            if strcmpi(varargin{i},'C'),
                C=varargin{i+1};
            end
            if strcmpi(varargin{i},'kerparam'),
                kerparam=varargin{i+1};
            end
        end
    end
end

if ~any(strcmpi(kerfunction,{'rbf','linear','poly'}))
    disp('Undefined kernel function.')
    disp('Type ''help svtutor_ctrain'' for more info')
    return
end
```

```
if C==inf,
    C = 10^6; % 'inf' is problematic in optimization using quadprog
end

% get number of samples in data matrix X
m = size(X,1);


K = svtutor_kernel(X,[],'kernel',kerfunction,'kerparam',kerparam);

% add a ridge to avoid ill-conditioned behaviour
Kreg = (y*y').*K + ridge*eye(size(K));

% set up optimization problem for MATLAB's quadprog
c = -ones(1,m); % linear factor in QP
A = zeros(1,m); % inequality constraints
neq = 0;
Aeq = y';
eq = 0;
LB = zeros(m,1);
UB = C*ones(m,1);

%call optimizer
x0 = []; %initial starting point
options = optimset('Display','off','LargeScale','off','MaxIter',maxIter);
[alphas,obj,exitflag,output,dual] = ...
quadprog(Kreg,c,A,neq,Aeq,eq,LB,UB,x0,options);
 bias = dual.eqlin(1);
%alphas = alphas*.y;
% get indices for SVs and non-bound SVs
iSVs = find(alphas>sv_cutoff);
nbSVs = find((alphas>sv_cutoff) & alphas<(C-sv_cutoff));

% compute weight vector
w = X(iSVs,:)'*(alphas(iSVs).*y(iSVs));


% assign to model
model.bias = bias;
model.SVs = X(iSVs,:);
model.iSVs = iSVs;
model.nbSVs =nbSVs;
model.alphas = alphas.*y;
model.w = w;
model.C = C;
```

```
model.kernel=kerfunction;
model.kerparam=kerparam;

return

%-------------------------------------------------------------------------

function Yt = svtutor_ctest(model,Xt)
% function svtutor_ctest(X,y,model)
%           Perform binary classification using a Support Vector
% Classifer (SVC) defined by the object model (obtained
%  from  svtutor_ctrain)
% INPUTS:
%             model - structure obtained from svtutor_ctrain defining the
%                         binary classification model
%             Xt  - test data
% OUTPUTS:
%            Yt - unsigned decision value function for Xt
%                  (Actual class is obtained by thresholding:  that is,
%                  Yact = sign(Yt);)
% DEPENDENCIES
%
%           svtutor_kernel
%
% REMARKS:
%               Only useful for demonstrations purposes for sample sizes
%               less than 500.
%               No input argument checking
%
% GT JEMWA, 2005
% Last Revision: 5 Feb, 2007

Kt = svtutor_kernel(model.SVs,Xt,'kernel',model.kernel,...
'kerparam',model.kerparam);
Yt = (model.alphas(model.iSVs)'*Kt+model.bias)';

%-------------------------------------------------------------------------
function K = svtutor_kernel(X,Xt,varargin)
% function svtutor_kernel(X,y,model)
%           Build a kernel matrix K using a kernel kerfunction with
%           parameters kerparam
%INPUTS:
%               X - data matrix
%               Xt - test data (if empty X is used)
%               kerfunction - kernel function to use
```

```
% ('linear' [default],'poly', 'rbf')
% See example below on calling function with
%               variable arguments
%               kerparam - kernel hyper parameter(s) associated with
%               kerfunction
% OUTPUTS:
%               K - kernel matrix
% DEPENDENCIES
%
% REMARKS:
%               Support linear, polynomial, and Gaussian kernels only
%               For tutorial purposes only.
%
% USAGE:
%               To create a linear kernel
%                  K=svtutor_kernel(X);
%
%               To create a polynomial function with degree 3:
%                  K=svtutor_kernel(X,[],'kernel','poly','kerparam',3)
%
%               To create a Gaussian function with width 0.5:
%                  K=svtutor_kernel(X,[],'kernel','rbf','kerparam',0.5)
%
%
% GT JEMWA, 2005
% Last Revision: 4 Feb, 2007
kernel = 'linear';
kerparam = 1; %default


for i = 1:length(varargin),
    if strcmp(varargin{i},'kernel'),
        kernel = varargin{i+1};
    elseif strcmp(varargin{i},'kerparam'),
        kerparam = varargin{i+1};
    end
end

%if strcmp(kernel
if isempty(Xt),
    Xt = X;
end

r1 = size(X,1);
r2  = size(Xt,1);
```

```
switch kernel
    case 'linear'
        K = X*Xt';
    case 'poly'
        K = (X*Xt' + 1).^kerparam;
    case 'rbf'
        dist2 = repmat((sum((X.^2), 2))', [r2 1])' ...
         + repmat((sum((Xt.^2),2))', [r1 1]) ...
         - 2*X*Xt';
        K = exp(-dist2/(2*kerparam^2));
end
```

B.2 Kernel Fisher Discriminant Analysis

```
% The following m-files (kfda.m,training.m,and testing.m) implement the
% nonlinear discriminant analysis using kernels. The files must be used
% as an object folder (that is, @kfda) in the folder clust of the Spider
% MATLAB machine learning environment (downloadable at
% http://www.kyb.tuebingen.mpg.de/bs/people/spider/)
% Note these extend the environment and not included in the currrent
% version (v1.7)
% USAGE: Given training data X and Y \in {1,2,3,...,M), where M is number
%  classes
%  1. Train an KFDA object, say using an RBF kernel of width 1 and
%  extracting 2 features;
%
% [res,model] = train(kfda({kernel('rbf',1),'feat=2'}),data(X,Y));
%
%  2. To project test data Xt on the model
%
% res = test(model,data(Xt));
%



%------------------------------------------------------------------------
function a = kfda(hyper)
%------------------------------------------------------------------------

%========================================================================
% KFDA kfda object - Kernel Fisher Discriminant Analysis
%========================================================================
% A=KFDA(H) returns a kpca object initialized with hyperparameters H.
%
```

```
% Hyperparameters, and their defaults
%  feat=0;            -- number of features, [default=2]
%  center_data=1;     -- if data is to be centered in feature space
%  child=linear       -- child stores the kernel. Default is the linear
%                        kernel and therefore normal lda.
% Model
%  e_val              -- the eigenvectors
%  e_vec              -- the eigenvalues
%  dat                -- training data (from which features where
%  extracted from)
%
% Methods:
%  train, test
% Implementation: GT JEMWA, 2006
%=========================================================================
% Reference : Generalized discriminant analysis using a  kernel approach.
%                 Neural Computation, 12,2000:2385-240
% Author    :  Baudat, G. and Anouar, A.
%=========================================================================

  %hyperparams
  a.feat=2;
  a.center_data = 1;
  a.child=kernel('linear');
  a.b0=0;

  % model
  a.e_vec=[]; % eigenvectors
  a.e_val=0;  % eigenvalues
  a.dat=[];
  a.Kt=[];
  p=algorithm('kfda');
  a= class(a,'kfda',p);

  if nargin==1,
    eval_hyper;
  end;


%-------------------------------------------------------------------------
function [results,a] =  training(a,d)
%-------------------------------------------------------------------------
% Implementation: GT JEMWA, 2006

% [results,algorithm] =  training(algorithm,data,loss)
```

```
disp(['training ' get_name(a) '.... '])


% sort data
X = get_x(d);
Y = get_y(d);
[dmp,ind] = sort(Y);
d = data(X(ind,:),Y(ind,:));



%calculate kernel
K=calc(a.child,d,[]);
a.Kt=K;
sumK = sum(K);
%center kernel in feature space
if (a.center_data)
    I=eye(length(K));
    O=ones(length(K))/length(K);
    K=(I-O)*K*(I-O);
end

%{
% compute rank first
if (a.feat == 0)
a.feat=rank(K);
end
%}
%decomposition of the centered matrix
[vecK,valK]=eig(K);
valK=real(diag(valK));
[dmp ind]=sort(-abs(valK));
valK=valK(ind);
rankK=length(find(valK>=valK(1)/1000));
valK=valK(1:rankK);
vecK=vecK(:,ind(1:rankK));

K=vecK*diag(valK)*vecK';

% build block diagonal matrix
groups=unique(d.Y);
blks=zeros(length(groups),1);
for i=1:length(groups),
    blks(i) = length(find(d.Y==groups(i)));
```

```
end

W=zeros(get_dim(d));
stBloc=1;
endBloc=0;
for i=1:length(groups)
    endBloc=endBloc+blks(i);
    for j=stBloc:endBloc
        for k=stBloc:endBloc
            W(j,k)=1/blks(i);
        end
    end
    stBloc=stBloc+blks(i);
end


% compute alpha normalized vectors % eigensystem%
K1 = vecK'*W*vecK;

if (a.feat < size(K1,1) - 1)
    opts.disp=0;
    [ a.e_vec, a.e_val]=eigs(K1, a.feat, 'LM', opts);
else
    [a.e_vec, a.e_val]=eig(K1);
end
a.e_val = real(diag(a.e_val));
%  a.feat=sum(a.e_val>1e-10);

% sort eigenvalues and eigenvector according to absolute size of
% eigenvals
[vals ind]=sort( -abs(a.e_val));
a.e_val=a.e_val( ind(1:a.feat));
a.e_vec=a.e_vec( :, ind(1:a.feat));
a.e_vec= vecK*diag(1./valK)*a.e_vec;

%normalize eigenvectors,
for i=1:a.feat,
    a.e_vec(:,i) = a.e_vec(:,i)/sqrt(a.e_vec(:,i)'*K*a.e_vec(:,i));
end


a.b0=(-sumK*a.e_vec/get_dim(d)+sum(a.e_vec)*sum(sumK)/get_dim(d)^2);
for i=1:a.feat,
    a.e_vec(:,i)=a.e_vec(:,i)-sum(a.e_vec(:,i))/get_dim(d);
end
```

```
a.dat=d;
results = test(a,d);

%-------------------------------------------------------------------------
function d =  testing( a, d)
%-------------------------------------------------------------------------
% Implementation: GT JEMWA, 2006

  K = calc(a.child, d, a.dat)'; %% between old examples and test examples
  if 0
    [Kt] = calc(a.child,a.dat,a.dat); %%restore old uncentered kernel
  else
    Kt=a.Kt;
  end

  if a.center_data   %center the test examples in feature space
    [n,m] = size(K);
    Om = ones(m)/m;
    On = ones(n,m)/m;
    I = eye( m);
    %brackets are for better numerical condition
    K = ((K - On*Kt)*(I-Om))';
  end

  test_features = K'*a.e_vec+ones(get_dim(d),1)*a.b0;
  %test_features = K'*a.e_vec;
  d=set_x(d,test_features);
  d.name=[get_name(d) ' -> ' get_name(a)];
```

B.3 One-class Support Vector Classification

```
%-------------------------------------------------------------------------
function net = train_1svm(x,KTYPE,KPAR,nu)
%-------------------------------------------------------------------------

% NET =  TRAIN_1SVM(X,KTYPE,KPAR,NU)
% 1-SVM for quantile estimation
% -------
% INPUTS
% -------
%          x - training data patterns
%          KTYPE (integer) - kernel type
%                        1 - linear: {default}
%                             x*x'
```

```
%                       2 - linear with bias:
%                            x*x' + b
%                       3 - polynomial of degree KPAR (X*X' + 1)^KPAR
%                            ((x * x') + 1)^KPAR
%                       4 - gaussian kernel
%                            exp(-1/(2*KPAR^2)* norm(x - y)^2)
%
%          KPAR - kernel hyperparamater corresponding to
%                    KTYPE. Thus, for a the linear, polynomial and gaussian
%                    kernels KPAR refers to the bias, degree, and kernel
%                    width. {1}
%          NU - specifies size of support region \equiv (1-NU)
% ---------
% OUTPUTS
% ---------
%        NET - a structure retaining the training parameters (X,KTYPE,KPAR,NU)
%  as well as the hyperplane weight vector coefficients (alphas)
% and offset (rho).

% author: gt jemwa 2006

m = size(x,1);
%H = sv_dot(rbf_dot(gamma),x',x');
H = compute_kernel(KTYPE,KPAR,x');
H = (H+H')./2;
H = H+1e-9*eye(size(H));

%setup qp optimization problem
c = zeros(1,m);
Aeq = ones(1,m);
eq = 1;
LB = zeros(m,1);
UB = (1/m/nu)*ones(m,1);

x0 = []; %initial starting point
options = optimset('Display','off','LargeScale','off','MaxIter',10000);
[alphas,obj,exitflag,output] = quadprog(H,c,[],[],Aeq,eq,LB,UB,x0,options);


% sanity check
if any(alphas>((1/m/nu)+1e-3)),
    error('something wrong...exiting');
end

nbsvi = (alphas>1e-6 & alphas <(1/m/nu)-1e-6);
```

```
%rho = H(nbsvi,nbsvi)*alphas(nbsvi,1);
% Kt = sv_dot(rbf_dot(gamma),x',x');
Kt = compute_kernel(KTYPE,KPAR,x');
yp = Kt*alphas;
rho = mean(yp(nbsvi));
net = struct('trainx',x,'KTYPE',KTYPE,'KPAR',KPAR,'nu',nu,'b0',rho,...
'alphas',alphas);

%-------------------------------------------------------------------------
function yt = test_1svm(net,xt)
%-------------------------------------------------------------------------

% YT = TEST_1SVM(NET,xt)
% -------
% INPUTS
% -------
%           NET - struct containing 1-SVM model (see TRAIN_1SVM.M)
%           xt  -  testing data
% ---------
% OUTPUTS
% ---------
%           YT - (non)-linear features
%
% Decision function: yt = (<w,x>-rho)
%                yt > 0  (point falling in estimated support region)
%                yt < 0  (points falling outside estimated support region,
%                                    "outliers")

% gt jemwa 2006
Kt = compute_kernel(net.KTYPE,net.KPAR,net.trainx',xt')';
yt = Kt*net.alphas - net.b0;

%-------------------------------------------------------------------------
function K=compute_kernel(KTYPE,KPAR,x1,x2)
%-------------------------------------------------------------------------


if nargin < 4,
    x2 = x1;
end

switch KTYPE
    case 1
        K = x1'*x2;
```

```
    case 2
        K = x1'*x2 + KPAR;
    case 3
        K=(x1'*x2+1).^KPAR;
    case 4
        dot_x1 = sum(x1.*x1,1);
        dot_x2 = sum(x2.*x2,1);

        unitvec = ones(size(x1,2),1);
        K = x1'*x2;
        for i=1:size(x2,2)
            K(:,i) = exp(1./(2*KPAR^2)* (2 * K(:,i) - dot_x1' - dot_x2(i)...
              * unitvec));
        end
    otherwise
        error('Unknown kernel function');
end
```

References

Aizerman, M., Braverman, E., and Rozonoer, L. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

Aldrich, C. and Barkhuizen, M. Process system identification strategies based on the use of singular spectrum analysis. *Minerals Engineering*, 16:815–826, 2003.

Aldrich, C., Moolman, D., Gouws, F., and Schmitz, G. Machine learning strategies for control of flotation plants. *Control Engineering Practice*, 5:263–269, 1997.

Aldrich, C. and Reuter, M. Monitoring of metallurgical reactors by use of topographic mapping of process data. *Minerals Engineering*, 12(11):1301–1312, 1999.

Aldrich, C., Roux, N.L., and Gardner, S. Monitoring of metallurgical process plants by use of biplots. *American Institution of Chemical Engineering Journal*, 50(6):2167–2186, 2004.

Aldrich, C. and Slater, M. Neural separation. In: G. Gardner (editor), *The Chemical Engineer*, vol. 586, page 11. The Institution of Chemical Engineers, 1995.

Aldrich, C. and Slater, M. Simulation of liquid-liquid extraction data with artificial neural networks. In: I. Mujtaba and M. Hussain (editors), *Application of Neural Networks and Other Learning Techniques in Process Engineering*, pages 3–22. Imperial College Press, 2001.

Aldrich, C. What is AI and is it better than classical process control? *South African Journal of Chemical Engineering*, 12(2):27–49, 2000.

Allen, M. and Smith, L. Distinguishing modulated oscillations from noise in multivariate datasets. *Climate Dynamics*, 12:775–784, 1996a.

Allen, M. and Smith, L. Monte Carlo SSA: detecting irregular oscillations in the presence of coloured noise. *Journal of Climate*, 9:3373–3404, 1996b.

Aradhye, H., Bakshi, B., Strauss, R., and Davis, J. Multiscale SPC using wavelets: Theoretical analysis and properties. *American Institution of Chemical Engineers Journal*, 49(4):939–958, 2003.

Aradhye, H., Davis, J., and Bakshi, B. ART-2 and multiscale ART-2 for on-line process fault detection – Validation via Monte Carlo simulation. *Annual Reviews in Control*, 26:113–127, 2002.

Aronszajn, N. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

Åström, K. and McAvoy, T. Intelligent control. *Journal of Process Control*, 2:115–127, 1992.

Ayoubi, M. and Isermann, R. Neuro-fuzzy systems for diagnosis. *Fuzzy Sets and Systems*, 89:289–307, 1997.

Bach, F. and Jordan, M. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

Bach, F., Lanckriet, G., and Jordan, M. Multiple kernel learning, conic duality, and the smo algorithm. In: *Proceedings of the 21st International Conference on Machine Learning*. Omnipress, Banff, Canada, 2004.

Bakir, G., Weston, J., and Schölkopf, B. Learning to find pre-images. In: S. Thrun, L. Saul, and B. Schölkopf (editors), *Advances in Neural Information Processing Systems*, vol. 16, pages 449–456. MIT Press, Cambridge, 2004.

Bakshi, B.R. Multiscale PCA with applications to multivariate statistical process monitoring. *American Institution of Chemical Engineers Journal*, 44(7):1596–1610, 1998.

Bakshi, B. and Stephanopoulos, G. Representation of process trends – IV. Induction of real-time patterns from operating data for diagnosis and supervisory control. *Computers and Chemical Engineering*, 18(4):303–332, 1994.

Barkhuizen, M. *Analysis of process data with singular spectrum methods*. Master's thesis, University of Stellenbosch, 2003.

Barnard, J., Aldrich, C., and Gerber, M. Identification of dynamic process systems with surrogate data methods. *American Institution of Chemical Engineers Journal*, 47(9):2064–2075, 2001.

Bartlett, P. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.

Basseville, M. and Nikiforov, I. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, Englewood Cliffs, N.J., 1993.

Baudat, G. and Anouar, A. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404, 2000.

Bishop, C. *Neural Networks for Pattern Recognition*. Oxford University Press, NY, 1995.

Blackman, R. and Tukey, J. *The Measurement of Power Spectra from the Point of View of Communication Engineering*. Dover, New York, NY, 1958.

Boger, Z. and Ben-Haim, M. Applications of neural networks techniques in solvent extraction: Modelling, knowledge acquisition and dynamic prediction. In: D. Logsdail and M. Slater (editors), *Proceedings ISEC '93, Solvent Extraction in the Process Industries*, pages 1198–1205. Elsevier Applied Science, 1992.

Boser, B., Guyon, I., and Vapnik, V. A training algorithm for optimal margin classifiers. In: D. Haussler (editor), *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992. Pittsburgh, PA, July 1992.

Boucheron, S., O., B., and Lugosi, G. Theory of classification: a survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

Bousquet, O. New approaches to statistical learning theory. *Annals of the Institute of Statistical Mathematics*, 55(2):371–389, 2003.

Box, G. and Kramer, T. Statistical process monitoring and feedback adjustment–a discussion. *Technometrics*, 34(3):251–267, 1992.

Bradley, P.S. *Mathematical programming approaches to machine learning and data mining*. Ph.D. thesis, University of Wisconsin - Madison, 1998.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. *Classification and Regression Trees*. Chapman & Hall, New York, NY, 1993.

Bro, R. PARAFAC: Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38:149–171, 1997.

Broomhead, D. and King, G. Extracting qualitative dynamics from experimental data. *Physica D*, 20:217–236, 1986.

Burges, C. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2):121–167, 2004.

Burges, C. Geometric methods for feature extraction and dimensional reduction – A guided tour. In: O. Maimon and L. Rokach (editors), *Data Mining and Knowledge Discovery Handbook*, pages 59–92. Springer, 2005.

Campbell, C. and Bennett, K. A linear programming approach to novelty detection. In: *Advances in Neural Information Processing Systems 14*, pages 395–401. MIT Press, Cambridge, 2001.

Carpenter, G.A. and Grossberg, S. ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, 3:129–152, 1990.

Cauwenberghs, G. and Poggio, T. Incremental and decremental support vector machine learning. In: *Advances in Neural Information Processing Systems*, vol. 13. MIT Press, 2001. Http://cbcl.mit.edu/projects/cbcl/publications/ps/cauwenberghs-nips00.pdf.

Cawley, G.C. MATLAB support vector machine toolbox (v0.55β) [`http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox`]. University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ, 2000.

Chen, J., Bandoni, J., and Romagnoli, J. Robust PCA and normal region in multivariate statistical process monitoring. *American Institution of Chemical Engineers Journal*, 42(12):3563–3566, 1996.

Chen, Q., Wynne, R., Goulding, P., and Sandoz, D. The application of principal component analysis and kernel density estimation to enhance process monitoring. *Control Engineering Practice*, 8:531–543, 2000.

Chiang, L.H., Kotanchek, Mark, E., and Kordon, Arthur, K. Fault diagnosis based on Fisher discriminant analysis and support vector machines. *Computers and Chemical Engineering*, 28:1389–1401, 2004.

Chiang, L.H. and Pell, R.J. Genetic algorithms combined with discriminant analysis for key variable identification. *Journal of Process Control*, 14:143–155, 2004.

Chiang, L., Russell, E., and Braatz, R. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principles component analysis. *Chemometrics and Intelligent Laboratory Systems*, 50:243–252, 2000.

Cho, J.H., Lee, J.M., Choi, S., Lee, D., and Lee, I.B. Fault identification for process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 60(1):279–288, 2005.

Choi, S., Changkyu, L., Lee, J.M., Park, L., and Lee, I.B. Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems*, 75:55–67, 2005.

Chouai, A., Cabassud, M., Le Lann, M., Gourdon, C., and Casamatta, G. Use of neural networks for liquid-liquid extraction column modelling: An experimental study. *Chemical Engineering and Processing*, 39:171–180, 2000.

Collins, M. and Duffy, N. Convolution kernels for natural language. In: T. Dietterich, S. Becker, and Z. Ghahramani (editors), *Advances in Neural Information Processing Systems*, vol. 14. MIT Press, 2002.

Cortes, C. and Vapnik, V. Support vector networks. *Machine Learning*, 20:273–297, 1995.

Cristianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, UK, 2000.

Dettinger, M., Ghil, M., and Kepenne, C. Interannual and interdecadal variability in United States surface-air temperatures, 1910–1987. *Climatic Change*, 31:35–66, 1995a.

Dettinger, M., Ghil, M., Strong, C., Weibul, W., and Yiou, P. Software expedites singular-spectrum analysis of noisy time series. *EOS Transactions of the American Geophysical Union*, 76(2):12,14,21, 1995b.

Diamantaras, K. and Kung, S. *Principal Component Neural Networks*. John Wiley & Sons, New York, NY, 1996.

Dong, D. and McAvoy, T. Nonlinear principal component analysis – Based on principal curves and neural networks. *Computers and Chemical Engineering*, 16:313–328, 1992.

Dreyfus, H. and Dreyfus, S. Making a mind versus modelling the brain: Artificial intelligence back at a brach-point. *Daedalus*, 117(1):15–43, 1988.

Dunia, R. and Qin, J. Joint diagnosis of process and sensor faults using principal control analysis. *Control Engineering Practice*, 6:457–469, 1998.

Dunia, R., Qin, J., Edgar, T., and McAvoy, T. Identification of faulty sensors using principal component analysis. *American Institution of Chemical Engineering Journal*, 42:3797–3812, 1996.

Elsner, J. and Tsonis, A. *Singular Spectrum Analysis - A new tool in time series analysis*. Plenum Press, New York, N.Y., 1996.

Everitt, B.S. and Dunn, G. *Applied Multivariate Data Analysis*. Oxford University Press, 2001.

Fisher, R. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

Fletcher, R. *Practical Methods of Optimization*. Wiley, New York, NY, 1989.

Frank, P.M. Fault diagnosis in dynamical systems using analytical and knowledge-based redundancy – A survey and some new results. *Automatica*, 26(3):459–474, 1990.

Frank, P., Ding, S., and Marcu, T. Model-based fault diagnosis in technical processes. *Transactions of the Institute of Measurement and Control*, 22(1):57–101, 2000.

Fukunaga, K. *Introduction To Statistical Pattern Recognition*. Academic Press, San Diego, 2nd ed., 1990.

Gabriel, K. The biplot graphical display of matrices with application to principal component analysis. *Biometrika*, 58:453–467, 1971.

Gardner, S., Le Roux, N., and Aldrich, C. Process data visualization with biplots. *Minerals Engineering*, 18:955–968, 2005.

Gärtner, T. A survey of kernels for structured data. *SIGKDD Explorations*, 5(1):49–58, 2003.

Ghil, M., Allen, M., Dettinger, M., Ide, K., Kondrashov, D., Mann, M., Robertson, A., Saunders, A., Tian, Y., Varadi, F., and Yiou, P. Advanced spectral methods for climatic times series. *Reviews of Geophysics*, 40(1):3.1–3.41, 2002.

Giles, A., Aldrich, C., and van Deventer, J. A modelling of rare earth solvent extraction with neural nets. *Hydrometallurgy*, 43(1–3):241–255, 1996.

Girolami, M. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 14:669–688, 2002.

Golyandina, N., Nekrutin, V., and Zhigljavsky, A. *Analysis of Time Series Structure: SSA and Related Techniques*. Chapman & Hall/CRC, Florida, USA, 2001.

Gomm, J., Weerasinghe, M., and Williams, D. Diagnosis of process faults with neural networks and principal component analysis. *Proceedings of the Institution of Mechanical Engineers. Part E, Journal of Process Mechanical Engineering*, 214:131–143, 2000.

Gower, J. and Hand, D. *Biplots*. Chapman & Hall, London, 1996.

Hastie, T. and Stuetzle, W. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York, NY, 2001.

Haussler, D. Convolutional kernels on discrete structures. Tech. Rep. Technical Report UCSC-CRL-99-10, Computer Science Department, UC Santa Cruz, 1999.

Haykin, S. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company, New York, NY, 1994.

Hayton, P., Schölkopf, B., Tarassenko, L., and Anuzis, P. Support vector novelty detection applied to jet engine vibration spectra. In: *Advances in Neural Information Processing Systems (NIPS)*, vol. 13, pages 946–952. 2000.

Hines, J. and Seibert, R. Technical review of on-line monitoring techniques for performance assessment. Vol. 1: State-of-the-art. Tech. rep., Division of Engineering Technology, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 2006. Available at: `http://www.nrc.gov/reading-rm/doc-collections/nuregs/`.

Hsieh, W. Nonlinear principal component analysis by neural networks. *Tellus*, 53A:599–615, 2001.

Hsieh, W. Nonlinear multivariate and time series analysis by neural network methods. *Reviews of Geophysics*, 42(1):RG1003.1–RG1003.25, 2004.

Hsieh, W. and Wu, A. Nonlinear multichannel singular spectrum analysis of the tropical pacific climate variability using a neural network approach. *Journal of Geophysical Research*, 107(C7):RG1003.,doi:10.1029/2002RG000112, 2002a.

Hsieh, W. and Wu, A. Nonlinear singular spectrum analysis. In: *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 3, pages 2819–2824. 2002b.

Hsu, C.W. and Lin, C.J. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425, 2002.

Hume, D. An enquiry concerning human understandings. In: *Essays and Treatises on Several Subjects*, vol. II, pages 5–165. Selb–Bigge, 1777.

Hyötyniemi, H. and Ylinen, R. Modeling of visual flotation froth data. *Control Engineering Practice*, 8:313–318, 2000.

Isermann, R. and Ballé, P. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719, 1997.

Isermann, R. Process fault detection based on modeling and estimation methods – A survey. *Automatica*, 20(4):387–404, 1984.

Isermann, R. Model-based fault-detection and diagnosis – status and applications. *Annual Reviews in Control*, 29:71–85, 2005.

Jackson, J.E. and Mudholkar, G. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3):341–349, 1979.

Jämsä-Jounela, S.L., Vermasvuori, M., Endé, P., and Haavisto, S. A process monitoring system based on the Kohonen self-organizing maps. *Control Engineering Practice*, 11:93–92, 2003.

Jia, F., Martin, E., and Morris, A. Nonlinear principal component analysis for process fault detection. *Computers and Chemical Engineering*, 22:S851–S854, 1998.

Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In: *Proceedings of the European Conference on Machine Learning*, pages 137–142. Springer, Berlin, 1998.

Jolliffe, I. *Principal Component Analysis*. Springer-Verlag, New York, NY, 2nd ed., 2002.

Joseph, B., Wang, F., and Shieh, D.S. Exploratory data analysis: A comparison of statistical methods with artificial neural networks. *Computers and Chemical Engineering*, 16(4):4, 1992.

Kashima, H., Tsuda, K., and Inokuchi, A. Kernels on graphs. In: K. Tsuda, B. Schölkopf, and J.P. Vert (editors), *Kernels and Bioinformatics*, pages 461–466. MIT press, Cambridge, MA, 2004.

Kepenne, C. An ENSO signal in soya bean futures prices. *Journal of Climate*, 8:1685–1689, 1995.

Kimeldorf, G. and Wahba, G. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

Kivinen, J., Smola, A., and Williamson, R. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2004.

Kohonen, T. *Self-Organizing Maps*. Springer-Verlag, Berlin, 1995.

Kourti, T., Lee, J., and MacGregor, J. Experiences with industrial applications of projection methods for multivariate statistical process control. *Computers and Chemical Engineering*, 20:S745–S750, 1996.

Kourti, T. and MacGregor, J. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28:3–21, 1995.

Kramer, M. Autoassociative neural networks. *Computers and Chemical Engineering*, 16:313–328, 1992.

Kramer, M. and Leonard, J. Diagnosis using backpropagation neural networks–analysis and criticism. *Computers and Chemical Engineering*, 14:1323–1338, 1990.

Kresta, J., MacGregor, J., and Marlin, T. Multivariate statistical monitoring of process operating performance. *Canadian Journal of Chemical Engineering*, 69:35–47, 1991.

Ku, W., Storer, R.H., and Georgakis, C. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30:179–196, 1995.

Kuss, M. *Nonlinear multivariate analysis with geodesic kernels*. Master's thesis, Technische Universität Berlin, 2002.

Kwok, J. and Tsang, I. The pre-image problem in kernel methods. *IEEE Transactions on Neural Networks*, 15(6):1517–1525, 2004.

Lynch, D. Chaotic behaviour of reaction systems: Parallel cubic autocatalators. *Chemical Engineering Science*, 47(2):347–355, 1992.

MacGregor, J. and Kourti, T. Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3):403–414, 1995.

MacGregor, J.F., Jaeckle, C., Kiparissides, C., and Koutodi, M. Process monitoring and diagnosis by multiblock PLS methods. *American Institution of Chemical Engineers Journal*, 40(5):826–838, 1994.

Mallat, S. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

Mangasarian, O. Mathematical programming in data mining. *Data Mining and Knowledge Discovery*, 42(1):183–201, 1997.

Markou, M. and Singh, S. Novelty detection: a review – part 1: statistical approaches. *Signal Processing*, 83:2481–2497, 2003.

Martin, E., Morris, A., and Zhang, J. Process performance monitoring using multivariate statistical process control. *Systems Engineering and Automation*, 143(2):132–144, 1996.

McAvoy, T. Intelligent "control" applications in the process industries. *Annual Reviews in Control*, 26:75–86, 2002.

Mika, K., Rätsch, G., Weston, J., Schölkopf, B., and Mürial, K.R. Fisher discriminant analysis with kernels. In: Y.H. Hu, E. Larsen, E. Wilson, and S. Douglas (editors), *Proceedings IEEE Neural Networks for Signal Processing Workshop*, pages 41–48. 1999.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A., and Müller, K.R. Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:623–628, 2003.

Miller, P., Swanson, R., and Heckler, C. Contribution plots: A missing link in multivariate quality control. *International Journal of Applied Mathematics and Computer Science*, 8(4):775–792, 1998.

Mineva, A. and Popivanov, D. Method for single-trial readiness potential identification,based on singular spectrum analysis. *Journal of Neuroscience Methods*, 98:91–98, 1996.

Minka, T. Automatic choice of dimensionality for PCA. In: *Advances in Neural Information Processing Systems, 13*, pages 598–604. 2001.

Minsky, M. and Papert, S. *Perceptrons*. MIT Press, Cambridge, MA, 1969.

Moolman, D., Aldrich, C., Schmitz, G., and van Deventer, J. The interrelationship between surface froth characteristics and industrial flotation performance. *Minerals Engineering*, 9(8):837–854, 1996.

Moolman, D., Aldrich, C., van Deventer, J., and Stange, W. The classification of froth structures in a copper flotation plant by means of a neural net. *International Journal of Mineral Processing*, 43:23–30, 1995.

Mukherjee, S. and Vapnik, V. Support vector method for multivariate density estimation. Tech. rep., A.I. Memo No. 1653, C.B.C.L. Paper No. 170, MIT AIT & CBCL, 1999. Available at: `ftp://ftp.ai-publications/1500-1999/AIM-1653.ps`.

Müller, K.R., Rätsch, S.M.G., Tsuda, K., and Schölkopf, B. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–202, 2001.

Negiz, A. and Çinar, A. Statistical monitoring of multivariable dynamic processes with state-space models. *American Institution of Chemical Engineers Journal*, 43(8):2002–2020, 1997.

Nomikos, P. and MacGregor, J. Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37(1):41–59, 1995.

Norvilas, A., Negiz, A., DeCicco, J., and Çinar, A. Intelligent process monitoring by interfacing knowledge-based systems and multivariate statistical monitoring. *Journal of Process Control*, 10:341–350, 2000.

Novikoff, A. On convergence proofs on perceptrons. In: *Proceedings of the Symposium on the Mathematical Theory of Automata*, vol. X11, pages 615–622. Polytechnic Institute of Brooklyn, 1962.

Ogunnaike, B. A contemporary industrial perspective on process control theory and practice. *Annual Reviews in Control*, 20:1–8, 1996.

Ormerod, P. and Campbell, M. Predictability and economic time series. In: C. Heij, J. Schuacher, B. Hanson, and C. Praagman (editors), *System Dynamics in Economic and Financial Models*. John Wiley, 1997.

Özyurt, B. and Kandek, A. A hybrid hierarchical neural network-fuzzy expert system approach to chemical process fault diagnosis. *Fuzzy Sets and Systems*, 83:11–25, 1996.

Patan, K. and Parisini, T. Identification of neural dynamic models for fault detection and isolation: the case of a real sugar evaporation process. *Journal of Process Control*, 15:67–79, 2005.

Patton, R., Uppal, F., and Lopez-Toribio, C. Soft computing approaches to fault diagnosis for dynamic systems: a survey. In: A. Edelmayer and C. Bànyàsz (editors), *Fault Detection, Supervision and Safety for Technical Processes*, vol. I. 2000.

Patton, R.J. Robustness in model-based fault diagnosis: The 1995 situation. *Annual Reviews in Control*, 21:103–123, 1997.

Peter He, Q., Joe Qin, S., and Wang, J. A new fault diagnosis method using fault directions in Fisher discriminant analysis. *American Institution of Chemical Engineers Journal*, 51(2):555–571, 2005.

Petti, T., Klein, J., and Dhurjati, P. Diagnostic model processor: Using deep knowledge for process fault diagnosis. *American Institution of Chemical Engineers Journal*, 36(4):565–575, 1990.

Pfeufer, T. and Ayoubi, M. Application of a hybrid neuro-fuzzy system to the fault diagnosis of an automotive electromechanical actuator. *Fuzzy Sets and Systems*, 89:351–360, 1997.

Platt, J.C. Sequential minimal optimization: A fast algorithm for training support vector machines. Tech. Rep. MSR-TR-98-14, Microsoft Research, 1998.

Platt, J., Cristianini, N., and Shawe-Taylor, J. Large margin DAGs for multiclass classification. In: S. Solla, T. Leen, and K.R. Müller (editors), *Advances in Neural Information Processing Systems 12*. MIT Press, Cambridge, MA, 2000.

Poggio, T. and Girosi, F. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

Poggio, T., Rifkin, R., Mukherjee, S., and Niyogi, P. General conditions for predictivity. *Nature*, 428:419–422, 2004.

Polonik, W. Minimum volume sets and generalized quantile processes. *Stochastic Processes and their Applications*, 69:1–24, 1997.

Popper, K. *The Logic of Scientific Discovery*. Harper Torch Books, New York, 2nd ed., 1968.

Prasad, P. and Davis, J. *An Introduction to Intelligent and Autonomous Control*, chap. A Framework for Knowledge-Based Diagnosis in Process Operations, pages 401–422. Kluwer Academic Publishers, 1992. Antsaklis, P.J. and Passino, K. M. (editors).

Quinlan, J. Induction of decision tress. *Machine Learning*, 1(1):81–106, 1986.

Quinlan, J. Decision trees and decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):339–346, 1990.

Raich, A. and Çinar, A. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *American Institution of Chemical Engineers Journal*, 42(4):995–1009, 1996.

Rasmussen, C.E. and Williams, C.K. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

Rätsch, G., Mika, S., Schölkopf, B., and Müller, K.R. Constructing boosting algorithms from SVMs: An application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1–16, 2002.

Rengaswamy, R., Mylaraswamy, D., Årzén, K.E., and Venkatasubramanian, V. A comparison of model-based and neural network-based diagnostic methods. *Engineering Applications of Artificial Intelligence*, 14:805–818, 2001.

Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1959.

Rousseeuw, P., Ruts, I., and Tukey, J. The bagplot: a bivariate boxplot. *The American Statistician*, 53:382–387, 1999.

Rousseeuw, P. and Van Driessen, K. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41:212–223, 1999.

Rozynski, G., Larson, M., and Pryszak, Z. Forced and self-organized shoreline response for a beach in the southern baltic sea determined through singular spectrum analysis. *Coastal Engineering*, 43(1):41–58, 2001.

Rumelhart, D., Hinton, G., and Williams, R. Learning representations by back-propagating errors. *Nature*, 323(9):533–536, 1986.

Saraiva, P. and Stephanopoulos, G. Continuous process improvement through inductive and analogical learning. *American Institution of Chemical Engineering Journal*, 33(2):161–183, 1992.

Schapire, R., Freund, Y., Bartlett, P., and Lee, W. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

Schoellhamer, D. Singular spectrum analysis for time series with missing data. *Geophysical Research Letters*, 28(16):3187–3190, 2001.

Schölkopf, B. *Support Vector Learning*. Ph.D. thesis, Technische Universität Berlin, 1997.

Schölkopf, B., Herbrich, R., Smola, A., and Williamson, R. A generalized representer theorem. Tech. Rep. NC2-TR-2000-81, NeuroCOLT, 2000a.

Schölkopf, B., Mika, S., Burges, C., Knirsch, P., Müller, K.R., Rätsch, G., and Smola, A. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.

Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., and R.C.Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

Schölkopf, B., Platt, J., and Smola, A. Kernel method for percentile feature extraction. Tech. rep., Microsoft Research, Microsoft Corporation, 2000b.

Schölkopf, B., Smola, A., Williamson, R., and Bartlett, P. New support vector algorithms. *Neural Computation*, 13(7):1443–1471, 2000.

Schölkopf, B. and Smola, A. *Learning with Kernels–Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.

Schölkopf, B., Smola, A., and Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

Schreiber, T. and Schmitz, A. Surrogate time series. *Physica D*, 142:346–382, 2000.

Shaw, A., Doyle III, F.J., and Schwaber, James, S. A dynamic neural network approach to nonlinear process modelling. *Computers and Chemical Engineering*, 21(4):371–385, 1997.

Silverman, B. *Density Estimation*. Chapman & Hall, London, 1986.

Small, M. and Judd, K. Correlation dimension: A pivotal statistic for non-constrained realizations of composite hypotheses in surrogate data analysis. *Physica D*, 120:389–400, 1998.

Smola, A. *Learning with kernels*. Ph.D. thesis, Technische Universität Berlin, 1998.

Smola, A., Bartlett, P., Schölkopf, B., and (editor), D.S. *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, 2000. Collection.

Smola, A.J. and Schökopf, B. Sparse greedy matrix approximation for machine learning. In: *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 911–918. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

Sorsa, T. and Koivo, H.N. Neural networks in process fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(4):815–825, 1991.

Sorsa, T. and Koivo, H.N. Application of artificial neural networks in process fault diagnosis. *Automatica*, 29(4):843–849, 1993.

Steinwart, I., Hush, D., and Scovel, C. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6:211–232, 2005. Available at: `www.jmlr.org`.

Stephanopolous, G. Emerging directions in computer applications to biotechnology: upgrading the information content of biological data. *Annual Reviews in Control*, 23:61–69, 1999.

Stephanopoulos, G. and Han, C. Intelligent systems in process engineering: A review. *Computers and Chemical Engineering*, 6/7:743–791, 1996.

Tax, D.M. *One-class Classification: Concept-learning in the Absence of Counter-examples*. Ph.D. thesis, Technische Universiteit Delft, 2001.

Tax, D. and Duin, R. Support vector domain description. *Pattern Recognition Letters*, 20:1191–1199, 1999.

Theiler, J. and Prichard, D. Constrained-realization Monte Carlo methods of hypothesis testing. *Physica D*, 94:221–235, 1996.

Theron, J. and Aldrich, C. Identification of nonlinearities in dynamic process systems. *Journal of the South African Institute of Mining and Metallurgy*, 104(3):191–200, 2004.

Thornhill, N. Finding the source of nonlinearity in a process with plant-wide oscillation. *IEEE Transactions on Control Systems Technology*, 3(13):434–443, 2005.

Tikhonov, A.N. and Arsenin, V.Y. *Solutions of Ill-posed Problems*. W. H. Winston, Washington, D.C., 1977.

Timmer, J. What can be inferred from surrogate data testing? *Physical Review Letters*, 85(12):2647, 2000.

Tipping, M. and Bishop, C. Probabilistic principal component analysis. 1997. Available at: `http://citeseer.ist.psu.edu/tipping99probabilistic.html`. Last Accessed: 24 November, 2006.

Tucker, W.T., Faltin, F.W., and Vander Wiel, S.A. Algorithmic statistical process control: an elaboration. *Technometrics*, 35(4):363–375, 1993.

Turing, A. Computing machinery and intelligence. *MIND LIX*, 2236:433–60, 1950.

Twining, C. and Taylor, C. The use of kernel principal component analysis to model data distributions. *Pattern Recognition*, 36:217–277, 2003.

Uraikul, V., Chan, C., and Tontiwachwuthikul, P. Artificial intelligence for monitoring and supervisory control of process systems. *Engineering Applications of Artificial Intelligence*, page DOI:10.1016/j.engappai.2006.07.002, 2006.

Utgoff, P. Incremental induction of decision trees. *Machine Learning*, 4:161–186, 1989.

Vander Wiel, S.A., Tucker, W.T., Faltin, F.W., and Doganaksoy, N. Algorithmic statistical process control: concepts and an application. *Technometrics*, 34(3):286–297, 1992.

Vapnik, V. *Estimation of Dependences Based on Empirical Data [in Russian]*. Nauka, Moscow, 1979. (English translation: Springer Verlag, New York, 1982).

Vapnik, V. *Statistical Learning Theory*. Wiley, New York, 1998.

Vapnik, V. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, second ed., 2000.

Vapnik, V. and Chervonenkis, A. A note on one class of perceptrons. *Automation and Remote Control*, 25:821–837, 1964.

Vapnik, V. and Chervonenkis, A. Uniform convergence of frequencies of occurrence of events to their probabilities. *Dokl. Akad. Nauk SSSR*, 181:915–918, 1968.

Vapnik, V. and Chervonenkis, A. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

Vapnik, V. and Chervonenkis, A. *Theory of Pattern Recognition [in Russian]*. Nauka, Moscow, 1974. (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung*, Akademie-Verlag, Berlin, 1979).

Vapnik, V. and Chervonenkis, A. Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability and its Applications*, 26(3):532–553, 1981.

Vapnik, V. and Chervonenkis, A. The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis*, 1(3):283–305, 1991.

Vapnik, V. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

Vautard, R. and Ghil, M. Singular spectrum spectrum analysis in nonlinear dynamics,with applications to paleoclimatic time series. *Physica D*, 35:395–424, 1989.

Vautard, R., Yiou, P., and Ghil, M. Singular-spectrum analysis: A toolkit for short, noisy chaotic signals. *Physica D*, 58:95–126, 1992.

Venkatasubramanian, V. and Rich, S. An object-oriented two-tier architecture for integrating compiled and deep-level knowledge for process diagnosis. *Computers and Chemical Engineering*, 12(9/10):903–921, 1988.

Venkatasubramanian, V., Vaidyanathan, R., and Yamamoto, Y. Process fault detection and diagnosis using neural networks – I. Steady-state processes. *Computers and Chemical Engineering*, 14(7):699–712, 1990.

Venkatasubramanian, V. Prognostic and diagnostic monitoring of complex systems for product lifecycle management: Challenges and opportunities. *Computers and Chemical Engineering*, 29:1253–1263, 2005.

Venkatasubramanian, V. and Chan, K. A neural network methodology for process fault diagnosis. *American Institution of Chemical Engineers Journal*, 35(12):1993–2002, 1989.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., and Yin, K. A review of process fault detection and diagnosis. Part III: Process history based methods. *Computers and Chemical Engineering*, 27:327–346, 2003.

Vert, J.P., Tsuda, K., and Schölkopf, B. A primer on kernel methods. In: B. Schölkopf, K. Tsuda, and J.P. Vert (editors), *Kernel Methods in Computational Biology*, pages 35–70. MIT Press, 2004.

Vert, R., Zwald, L., Blanchard, G., and Massart, P. Kernel Projection Machine: a new tool for pattern recognition. In: *Proceedings of the 18th. Conference on Neural Information Processing Systems (NIPS 2004)*. 2005.

Vert, R. *Theoretical Insights On Density Level Set Estimation, Application to Anomaly Detection*. Ph.D. thesis, Université Paris XI – Paris Sud U.F.R. Scientifique D'orsay, 2006.

Vert, R. and Vert, J.P. Consistency and convergence rates of one-class svms and related algorithms. *Journal of Machine Learning Research*, 7:817–854, 2006. Available at: `www.jmlr.org`.

Watanabe, K., Hirota, S., Hou, L., and Himmelblau, D. Diagnosis of multiple simultaneous faults via hierarchical artificial neural networks. *American Institution of Chemical Engineers Journal*, 40(5):839–848, 1994.

Watkins, C. Dynamic alignment kernels. In: A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans (editors), *Advances in Large Margin Classifiers*, pages 39–50. MIT Press, Cambridge, MA, 2000.

Westcott, J. Discussion of "Some statistical aspects of adaptive optimization and control," by G.E.P. Box and G.M. Jenkins. *Journal of the Royal Statistical Society B*, 24:340–341, 1962.

Williamson, R.C., Smola, A.J., and Schölkopf, B. Generalization bounds for regularization networks and support vector machines via entropy numbers of compact operators. *IEEE Transactions on Information Theory*, 47(6):2516–2532, 2001.

Wise, B. and Gallagher, N. The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6):329–348, 1996.

Woinaroschy, A. Use of neural nets for dynamic simulation of liquid-liquid extraction. *Hungarian Journal of Industrial Chemistry*, 26(2):121–123, 1998.

Wolpert, D.H. and Macready, W.G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67–82, 1997.

Yoon, S. and MacGregor, J. Statistical and causal model-based approaches to fault detection and isolation. *American Institution of Chemical Engineers Journal*, 46(9):1813–1824, 2000.

Yoon, S. and MacGregor, J.F. Principal component analysis of multiscale data for process monitoring and fault diagnosis. *American Institution of Chemical Engineers Journal*, 50(11):2891–2903, 2004.